

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**SDK de recolección y plataforma de visualización enfocado al
marketing digital**

Alejandro Velasco García
Tutor: Germán Montoro

Junio 2020

SDK de recolección y plataforma de visualización enfocado al marketing digital

AUTOR: Alejandro Velasco García

TUTOR: Germán Montoro

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2020**

Resumen

La analítica enfocada al marketing son los cimientos para conocer en detalle como funciona una página web y como interactúan los usuarios con ella permitiendo mejorar características como su rendimiento, usabilidad y experiencia de usuario.

La analítica en medios digitales no es ni mucho menos un área reciente, si no que lleva practicándose desde los años '90, sin embargo, no fue hasta hace unos años con la transformación digital de muchas empresas dedicadas a la información y venta de productos principalmente cuando experimento un gran auge, y por ello encontramos ahora mismo que todavía es un área algo desconocida, pero con un prometedor futuro por delante.

Este proyecto ha tratado de recrear el proceso completo de implementación de un sistema de analítica web siguiendo un esquema propio desde la definición de los datos que queremos medir hasta su visualización en diversas herramientas. Para ello se ha utilizado una web de pruebas similar a una tienda online de ropa que cumple con las funcionalidades básicas (<https://tfgdashboard.de//ecommerceTest/menu.php>) pero que sirve perfectamente para mostrar todo el proceso.

Además de recrear el proceso completo, hay que añadir el desarrollo de un SDK capaz de implantarse en cualquier web utilizando un tag manager, y añadiéndole una configuración mínima ser capaz de recolectar información de una web y enviarla a diversas herramientas para el análisis y la visualización de los mismos. Cabe destacar también que este SDK se ha intentado hacer lo más sencillo y a la vez genérico posible buscando desarrollar una solución lo más estándar posible, escalable y mantenible.

Por último, se ha desarrollado una herramienta que permite el análisis y la visualización de los datos recolectados por el SDK.

Abstract

Analytics focused in marketing are the base for knowing in detail how a web page works and how users interact with it, improving features such as performance, usability and user experience.

Analytics in digital media is not a new area, it has been practicing since the 90s, however, it was not until a few years ago with the digital transformation of many companies dedicated to the information and sale of products mainly when it experiences a great increase, and that's why we find right now that it is still a somewhat unknown area, but with a promising future ahead.

This project has tried to recreate the complete process of implementing a web analysis system following its own scheme, from the definition of the data we want to measure to display data in different tools. For this, a test website similar to an online clothing store that complies with the basic functionalities (<https://tfgdashboard.de//ecommerceTest/menu.php>) has been used, but it serves perfectly to show the whole process.

In addition to recreating the entire process, it has been developed a SDK capable of implementing on any website using a tag manager, and add a minimal configuration transfer that is capable of collecting information from a website and sending it to various tools to their analysis and visualization. It should also be noted that this SDK has tried to make it as simple and generic as possible, seeking to develop a solution that is as standard as possible, scalable and maintainable.

Finally, a tool has been developed that allows the analysis and visualization of the data collected by the SDK

Palabras clave

- **Analítica web:** la medición, procesamiento, análisis y reporte del tráfico de internet para entender y optimizar el uso de un sitio web
- **Core avcs:** SDK desarrollado que permite extraer información de páginas web y enviarla a las herramientas de analítica configuradas. También incluye diversas funcionalidades para el tratamiento de la información extraída.
- **Evento:** son interacciones del usuario con contenido que se pueden medir de forma independiente a partir de una página web o una carga de pantalla
- **GA:** es un servicio gratuito prestado por Google que ofrece un gran conjunto de estadísticas de tu página/s web/s
- **GTM:** es el tag manager desarrollado por Google.
- **Marketing digital:** son todas aquellas acciones y estrategias publicitarias o comerciales que se ejecutan en los medios y canales de internet.
- **Pageview/Página vista:** número de páginas vistas que se cargan (o se vuelven a cargar) en un navegador.
- **SDK (software development kit):** es un paquete de herramientas y datos que facilita e incluso permite a los programadores desarrollar programas en un lenguaje concreto o para una plataforma o aplicación específica.
- **Sesión:** una sesión es un conjunto de interacciones que tienen lugar en su sitio web desde que se accede a la web hasta que se cierra la web o el navegador.
- **Tag/Etiqueta:** término utilizado para referirse en Google Tag Manager a segmentos de código proporcionados por proveedores de analítica o marketing para integrar los servicios que ofrecen.
- **Tag manager:** es un sistema de administración o gestor de etiquetas que nos permite actualizar de forma fácil y rápida todas aquellas etiquetas/píxeles/códigos dentro de su plataforma sin tener que añadir directamente el código en la web o apps.
- **TFGdashboard:** herramienta desarrollada durante este proyecto para visualizar y analizar los datos en enviados desde el core avcs.
- **Trigger/Activador:** se aplican en Google Tag Manager para detectar ciertos eventos y poder ejecutar servicios a través de etiquetas.

Keywords

- Web analytics: the measurement, processing, analysis and reporting of internet traffic to understand and optimize the use of a website
- Core avcs: SDK developed that allows information to be extracted from web pages and sent to the configured analytical tools. It also includes various functionalities for the treatment of the extracted information.
- Event: user interactions with content that can be measured independently from a web page or a screen load.
- GA: is a free service provided by Google that offers a large set of statistics for your web page / s.
- GTM: is the tag manager developed by Google.
- Digital Marketing: all those actions and advertising or commercial strategies that are executed in the media and internet channels.
- Pageview / Page view: number of page views that are loaded (or reloaded) in a browser for a web page.
- SDK (software development kit): it is a package of tools and data that facilitates and even allows programmers to develop programs in a specific language or for a specific platform or application.
- Session: is a set of interactions that take place on your website from when the web is accessed until the web or browser is closed.
- Tag / Label: term used in Google Tag Manager to refer to code segments provided by analytics or marketing providers to integrate the services they offer.
- Tag manager: it is an administration system or tag manager that allows us to update easily and quickly all those tags / pixels / codes within your platform without having to directly add the code to the web or apps.
- TFGdashboard: tool developed during this project to visualize and analyze the data sent from the core avcs.
- Trigger / Activator: it is applied in Google Tag Manager to detect certain events and to be able to execute services through tags.

- ***Agradecimientos***

Quiero agradecer este trabajo a todos los profesores que me han dado clase durante estos años en los que he pasado por la universidad, a mi tutor Germán Montoro por darme la oportunidad de presentarme y ayudarme a elaborar mi trabajo, y a mis compañeros que me han ayudado durante estos años para poder llegar hasta aquí.

También quiero agradecer a mis compañeros de trabajo Manu, Miriam, Juli y Alex por enseñarme todo lo que sé sobre analítica digital y hacer del trabajo algo divertido, y en especial a Iván por confiar en mí y ser quién me dio la idea y me ayudó a comenzar este proyecto que hace unos meses parecía inalcanzable.

Y por último a mi familia por apoyarme siempre y no dejar que me rindiera.

Gracias a todos.

INDICE DE CONTENIDOS

1	Introducción.....	11
1.1	Motivación.....	11
1.2	Objetivos.....	12
2	Estado del arte.....	13
2.1	Estado del arte.....	13
3	Diseño.....	15
3.1	Web de muestra.....	15
3.1.1	Menú.....	15
3.1.2	Producto.....	15
3.1.3	Carrito.....	16
3.1.4	Shipping.....	16
3.1.5	Pago.....	16
3.1.6	Confirmación de compra.....	17
3.1	Documento de recolección.....	17
3.2	Aves.....	17
3.3	Google Analytics.....	18
3.4	TFGdashboard.....	18
3.4.1	General.....	18
3.4.2	Diseño web.....	19
3.4.3	Identificación.....	24
3.4.4	Base de datos.....	24
3.4.5	Gestión.....	24
3.4.6	Gráficos.....	24
3.4.7	Canales.....	25
4	Desarrollo.....	27
4.1	Documento de recolección.....	27
4.1.1	Índice.....	27
4.1.2	Encabezado.....	27
4.2	Web de muestra.....	28
4.3	SDK (core avcs).....	29
4.3.1	Config.....	29
4.3.2	Ecommerce.....	29
4.3.3	Helpers.....	30
4.3.4	event.....	31
4.3.5	printOut.....	31
4.3.6	init.....	32
4.3.7	send.....	32
4.3.8	push.....	32
4.3.9	vars.....	33
4.3.10	setCustomVars.....	36
4.4	Google Tag Manager.....	36
4.4.1	Seguimiento.....	36
4.4.2	Cookie.....	36
4.4.3	Nuevos usuarios.....	37
4.4.4	Implementación.....	37
4.5	Google Analytics.....	41
4.6	TFGdashboard.....	41
4.6.1	Servidor.....	41
4.6.2	Base de datos.....	42

4.6.3 Gráficos.....	43
4.6.4 Canales.....	44
4.6.5 Filtros.....	45
4.6.1 Implementación.....	46
5 Integración, pruebas y resultados.....	49
5.1 Pruebas en página.....	49
5.2 Pruebas en Google Tag Manager.....	49
5.3 Pruebas en informes.....	49
6 Conclusiones y trabajo futuro.....	50
6.1 Conclusiones.....	50
6.2 Trabajo futuro.....	51
Referencias.....	53
Glosario.....	55
Anexos.....	I
A Manual de instalación.....	I
B Documento de recolección para páginas.....	- 1 -
D Documento de recolección para eventos.....	- 2 -
E Documento de recolección para eventos de ecommerce.....	- 4 -
F Documento de recolección de eventos de eventos para GA.....	- 5 -

1 Introducción

1.1 Motivación

La idea de este trabajo surgió hace casi 2 años, en 2018, cuando comencé mis prácticas extracurriculares en la empresa DBi que se dedicaba al marketing digital, y más concretamente, mi equipo se dedicaba principalmente a la recolección de datos en páginas web y su exportación a herramientas de analítica entre otras muchas tareas. Este proceso de recolección y envío de datos puede ser bastante complejo en muchos casos ya que influyen infinidad de variables como la velocidad de la carga de página, el dispositivo, las extensiones del navegador, ... así que en muchos casos puede darse que la implementación de la recolección de datos esté a priori correcta pero cualquiera de estas variables podían hacer que parte de la medición se cayera, lo cual es algo muy interesante ya que al final cada página es un problema distinto y pequeños cambios en su implementación pueden suponer desafíos enormes de cara a la medición. Al final todos estos obstáculos que surgen a la hora de recolectar los datos descentralizan el código y la implementación en el tag manager haciendo los contenedores difícilmente mantenibles y poco escalables. Por esta razón surgió la idea del core avcs, un único código JavaScript que junta todo lo necesario para recolectar los datos de una página web ya que posee una estructura de variables propia, un módulo para el tratamiento de datos, funciones de ayuda para simplificar procesos y una configuración para enviar los datos a las principales herramientas de analítica. Se ha buscado diferenciar bien los módulos auxiliares de los del proceso de envío para hacer que sea un código lo más sencillo posible, pero a la vez claro para que se pueda entender con facilidad y sea que se entienda bien haciendo que su mantenimiento no suponga ningún problema. Otra de las ideas que se buscaba con esta estructura de código era que fuera fácilmente extensible, es decir, que cada programador pueda hacer su aportación al código de manera limpia para añadir funcionalidades útiles y accesibles siempre.

Con la parte de la recolección cubierta el siguiente paso es saber por qué se quieren esos datos y que pueden aportar a un negocio. Para ello surgió la idea de diseñar un flujo para llevar los datos desde la recolección a su visualización y la herramienta TFGdashboard cumple con la tarea de visualización donde ver el comportamiento en la página web, estudiarlo y permitir tomar decisiones en base a estos datos para mejorar el rendimiento de la web y superar los objetivos.

1.2 Objetivos

A continuación, se enumeran los objetivos perseguidos con este trabajo:

1. Comprender mejor el flujo desde la recolección de datos en una página web hasta la consolidación de estos datos en las herramientas de analítica.
2. Mejorar las técnicas de recolección de datos.
3. Comprender por qué son necesarios los datos en el estudio de una página web
4. Aprender a interpretar datos en una herramienta de analítica.
5. Ponerme en el lado de los desarrolladores para tener una visión completa de los problemas de la web.
6. Descubrir nuevas herramientas y metodologías que puedan ser útiles para la analítica y mejorar en otras ya conocidas:
 - a. Firebase
 - b. Google Analytics
 - c. Google Tag Manager
 - d. Local/Session storage
7. Consolidar y perfeccionar el core avcs.
8. Descubrir limitaciones en las herramientas de analítica y como podrían solucionarse
9. Conocer otras herramientas y mejorar otros aspectos fuera de la analítica:
 - a. Protocolo FTP
 - b. Consultas SQL
 - c. PHP y html
 - d. Trabajar con un servidor

2 Estado del arte

2.1 Estado del arte

Actualmente muchas de las implementaciones en Google Tag Manager [2] para la medición de páginas web que existen son demasiado estáticas, es decir, utilizan un tag y trigger por evento o grupo de eventos ocurridos en la página haciendo que los desarrollos y el mantenimiento de estas soluciones sean costosas, poco eficientes y poco escalables, ya que cada evento añadido requeriría de su implementación en página y la creación de un tag y trigger con toda su configuración.

Sin embargo, cada vez surgen más soluciones que intentan dotar de un mayor dinamismo y eficiencia a estas soluciones intentando centralizar los eventos de las páginas. Esto se consigue agrupando lo máximo posible los eventos por categorías para evitar tener un tag y un trigger por cada evento, si no que por cada grupo de eventos que compartan ciertas características en su configuración de Google Tag Manager se pueda crear un solo tag y un solo trigger. De este modo los valores se cogen directamente de la llamada al tag manager desde página dotando a las variables de dinamismo en sus valores y haciendo que al tener menos tags y triggers sea una solución más mantenible y aumentando su eficiencia.

Desde mi punto de vista estos grupos de eventos podrían ser solamente dos, páginas vistas y eventos genéricos. La idea es que se activen los tags de páginas vistas únicamente en la carga de cada página, y los eventos genéricos con cualquier otro evento que se invoque, y sus valores serían totalmente dinámicos ya que se alimentarían ambos tags únicamente de variables informadas en el dataLayer.

Otras soluciones implementadas que se utilizan habitualmente en páginas web son objetos JavaScript auxiliares creados por los desarrolladores web para informar de los valores iniciales (previamente acordados con los equipos de analítica) de la página que se obtienen de un proveedor externo al que hay que hacer peticiones como puede ser la información de productos. Gracias a este objeto desde tag manager se tiene acceso a información como la tipología de la página o el nombre de un producto que puede asignarse a variables en Google Tag Manager.

Desde mi solución se propone eliminar este objeto y utilizar uno propio (**avcs**) donde se asignen los valores y se pueda tener controlado en todo momento el nombre de las variables para facilitar su acceso y de esta forma centralizar la información. Incluso de no existir en un primer momento esta variable auxiliar se podrían recoger los datos con librerías como jQuery o utilizar JavaScript, pero lo más importante es que se estaría utilizando un solo objeto y que las variables estarían predefinidas por lo que se conoce en todo momento cuales existe y como acceder a ellas.

También se ha tenido en cuenta el módulo de Enhanced Ecommerce de Google Analytics [1], una parte fundamental de la medición para webs que se dedican a la venta de productos online.

Por otro lado, otro de los tag managers más importantes del mercado es Adobe Launch, que es la herramienta ofrecida por Adobe pero que a diferencia de Google Tag Manager no ofrece una versión gratuita por lo que es más difícil de poder trabajar, sin embargo, he podido acceder a algunos cursos para conocer mejor su funcionamiento y utilizar algunas de sus ventajas.

Adobe Launch ofrece en su implementación un objeto JavaScript que permite tener acceso a diferentes funcionalidades y módulos desde la página como por ejemplo funciones para crear cookies.

Este aspecto es importante a la hora de facilitar la recolección de los datos, por lo que al objeto **avcs** se ha añadido un módulo con funciones auxiliares para la poder llevar a cabo diversas acciones como crear cookies, crear fechas o imprimir mensajes.

Además, una de las grandes ventajas de esta solución es su escalabilidad y flexibilidad para adaptarse a cualquier página sin necesitar de grandes desarrollos por lo que cada implementación puede personalizar el objeto **avcs** a según sus necesidades para recoger nuevas variables, utilizar nuevas funcionalidades o incluso añadir un módulo para el envío a nuevas herramientas de analítica.

Por último, se ha desarrollado la plataforma TFGdashboard como una herramienta para la visualización y estudio de los datos recogidos por **avcs**. Esta plataforma se basa principalmente en Google Tag Manager y Data Studio, dos herramientas que permiten visualizar datos recogidos en página, hacer estudios avanzados de los mismos para mejorar el rendimiento de la web, y en el caso de Data Studio elaborar informes para presentaciones.

TFGdashboard no posee la potencia y el alcance de ambas herramientas, sin embargo se sirve de muchas de sus propuestas para poder elaborar informes básicos y ayudar a comprender cuales son las limitaciones de este tipo de plataformas y que tipo de estudios podrían hacerse.

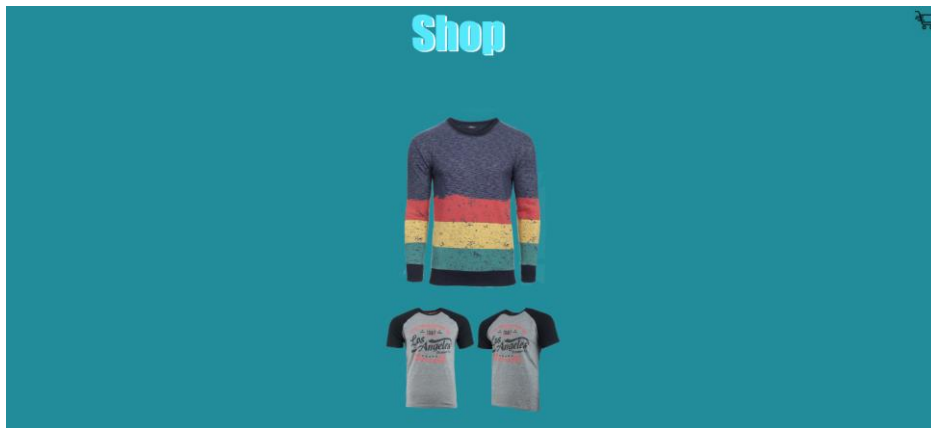
3 Diseño

En este apartado se van a explicar las decisiones de diseño y la implementación de cada una de las partes que conforman el auditado, el SDK y la plataforma TFGdashboard.

3.1 Web de muestra

Para poder probar el funcionamiento de esta infraestructura se ha utilizado una web muy básica que simula el funcionamiento de una tienda online de ropa y que cumple con todas las principales funcionalidades para mostrar cómo se desarrollaría en web real (<https://tfgdashboard.de/ecommerceTest/menu.php>). La idea es ver como si una web cumple con las funcionalidades puede implementar todo el sistema para la medición incluso en un estado muy básico. La estructura de la web es la siguiente:

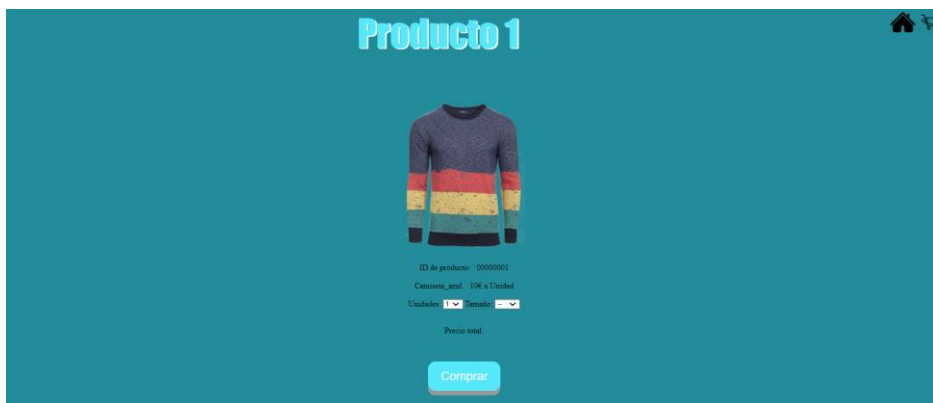
3.1.1 Menú



3.1.1.1 Web de muestra - Menú

En esta página se muestra una lista con todos los productos disponibles para la compra. Cuando se pulsa en un producto se accede a la página del producto.

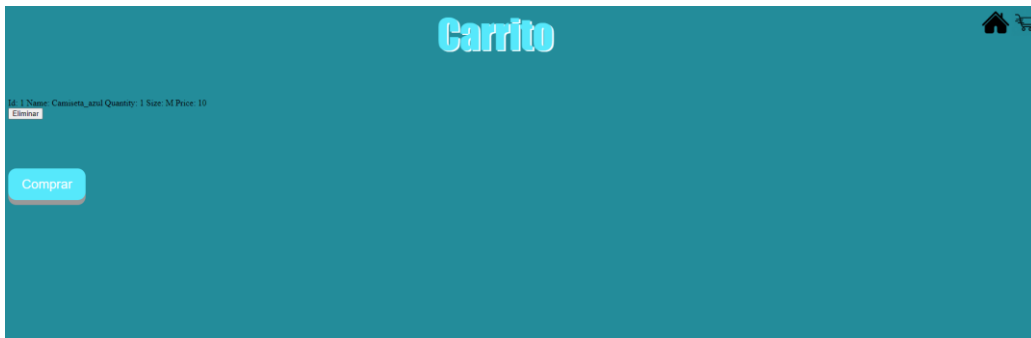
3.1.2 Producto



3.1.1.2 Web de muestra – Producto en detalle

Desde esta página se pueden ver los datos de un producto en detalle y además añadir al carrito de compra seleccionando la talla deseada.

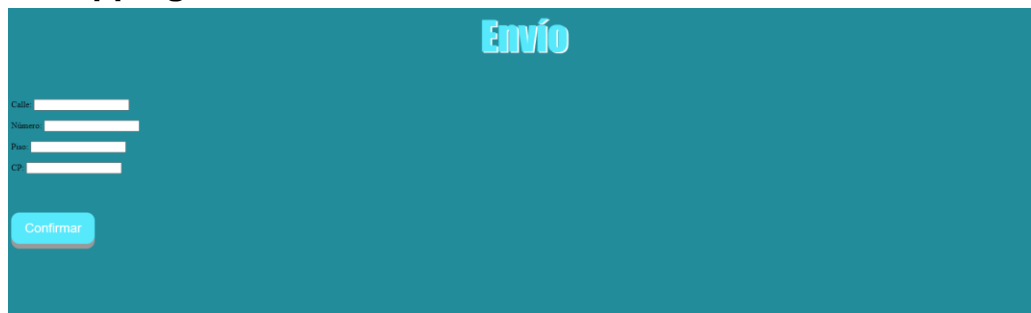
3.1.3 Carrito



3.1.1.3 Web de muestra – Carrito de compra

En la página de carrito se puede ver el estado actual del carrito de compra y si se pulsa en Comprar se accede al primer paso del proceso de compra.

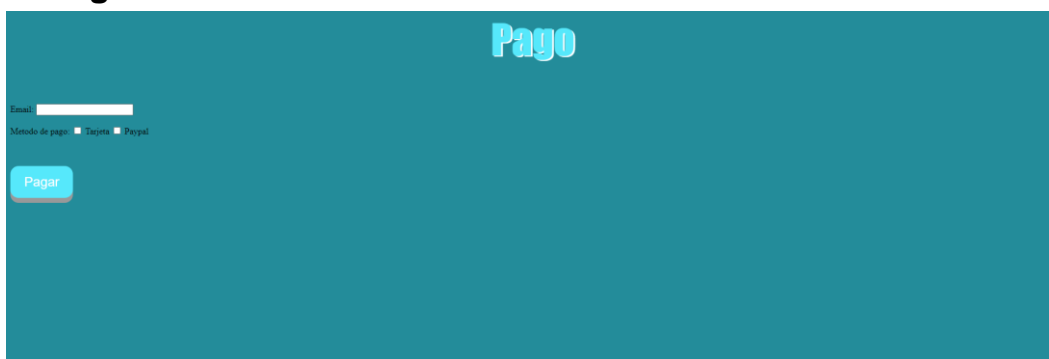
3.1.4 Shipping



3.1.1.1 Web de muestra – Proceso de pago – Paso de envío

Esta página pertenece al primer paso del proceso de compra donde se deberán rellenar los datos del envío del producto y pulsar en confirmar para vanzar al siguiente paso.

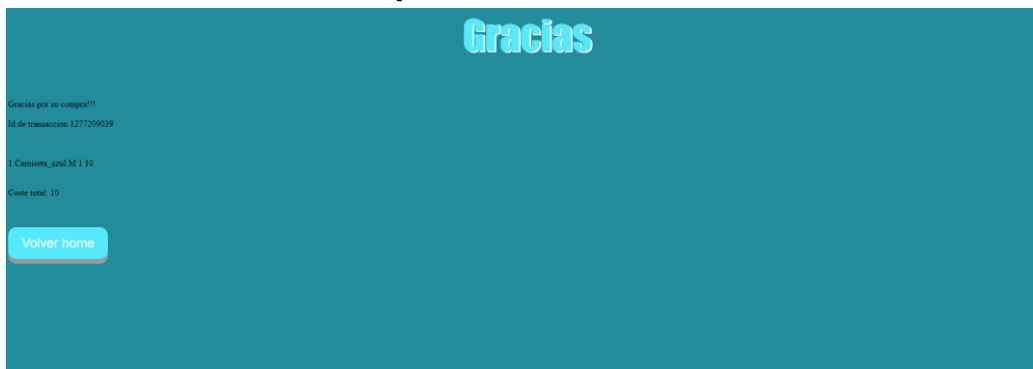
3.1.5 Pago



3.1.1.1 Web de muestra – Proceso de pago – Paso de pago

Esta página pertenece al segundo paso del proceso de compra donde se especificarán los datos para el pago del carrito y pulsando Pagar se confirmará la compra para acceder al último paso de la compra.

3.1.6 Confirmación de compra



3.1.1.6 Web de muestra – Confirmación de compra

Esta página ya confirma que se ha terminado el proceso de compra e informa de los datos finales de la compra. Por último, pulsando Volver home se retrocede a la página de home.

3.1 Documento de recolección

Para empezar a recolectar datos de una página lo primero es definir que se va a medir y que datos interesa conocer ya que no todas las páginas permiten recoger los mismos datos y dependiendo de la tipología de la web pueden resultar algunos datos más interesantes que otros. Por estas razones es importante completar el **documento de recolección** y definir qué datos se requieren y en qué momentos.

Este documento trata de plasmar la solución de recolección implementada en la página para que en cualquier momento haya una guía o una referencia sobre la solución la que acudir.

Su desarrollo debe ser el primer paso del proyecto por delante de la implementación de la medición ya que el objetivo es que los valores que se incluyan en este documento sean fijos y no se modifican salvo excepciones para que tanto los desarrolladores como el equipo de analítica tengan claros los valores que se deben informar en cada página y evento para poder implementar la solución correcta y posteriormente validarla.

El documento contendrá todas las páginas y eventos definidos para la página web y el valor que toman cada una de las variables existentes, teniendo así una vista general de todos los casos de páginas y eventos que pueden darse y las variables y valores que deberían informarse.

3.2 Avcs

Para llevar a cabo la recolección y gestión de los datos en página y enviar a herramientas de analítica se utilizará Google Tag Manager para implementar el SDK core avcs que cuya función principal es construir objeto JavaScript llamado **avcs** que se encargará de realizar todos estos procesos.

Este objeto debe ser la representación del documento de recolección en página y pretende centralizar toda la solución propuesta para la recolección y tratamiento de datos y el envío de los mismos a las herramientas de analítica (Anexo G).

Será un objeto independiente de las soluciones implementadas en la página ya que únicamente se le deberá informar de las variables recogidas, el tipo evento que mide y las herramientas de analítica a las que queremos enviar los datos.

Esta estructura del objeto **avcs** se ha llevado a diseñado en base a mi experiencia a otras implementaciones y como cliente en otros servicios web. El objetivo es establecer una estructura lo más general posible pero que a la vez no perdiera información de recolección

y se centrará en las páginas de ecommerce ya que considero que son las que más interesadas podrían estar en implementar esta solución.

Este diseño no tiene que porque ser válido para todas las páginas, aunque se ha intentado que abarque la mayor cantidad de campos básicos y no debería distar mucho de las soluciones finales, pero al final cada página y cada implementación es única por lo que es muy posible que dependiendo la web hubiera que añadir alguna variable más.

Cada módulo cumple con una funcionalidad para apoyar las recolección y tratamiento de los datos.

3.3 Google Analytics

Para la visualización de datos se dispone de Google Analytics [7], una herramienta de analítica web desarrollada por Google y que permite visualizar y analizar datos de nuestra página web utilizando multitud de informes y gráficos personalizables.

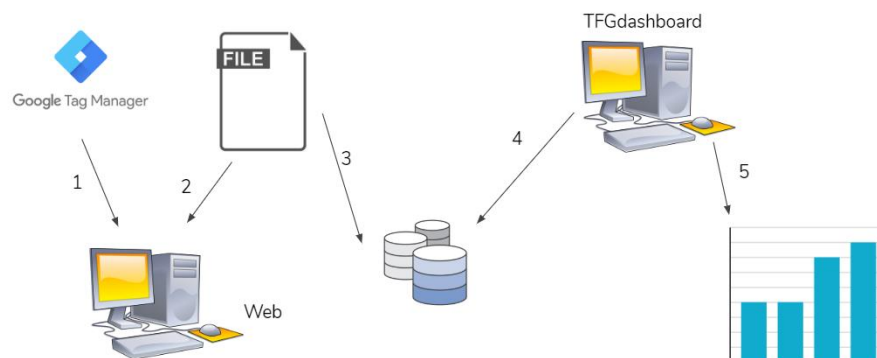
Se decidió utilizar esta Google Analytics para la visualización porque es la principal herramienta del mercado al facilitar un buen nivel de informes para la visualización y la posibilidad de interconectarla con otras herramientas de Google como Data Studio, además de ser de las pocas herramientas con una versión gratuita.

3.4 TFGdashboard

Se ha creado la herramienta TFGdashboard (<https://tfghashboard.de/tfghashboard/inicio.php>) con el objetivo de comprender y tener una visión más completa de cómo es el flujo entre la recolección de datos entre y su visualización, y tener una plataforma alternativa a Google Analytics para estudiar los datos recolectados y conocer cómo funcionan este tipo de plataformas, sus limitaciones y como podrían mejorarse. A continuación, se van a exponer los diseños que se han llevado a cabo para desarrollar la plataforma.

3.4.1 General

El siguiente esquema muestra el flujo de los datos:



3.6.1.1 TFGdashboard – Esquema general

1. El core avcs debe cargarse en la página web de la que se van a recoger los datos a través de Google Tag Manager.
2. Se debe implementar la función `sendTo()` que se encargará de enviar los datos recolectados a la base de datos.
3. La función `sendTo()` enviará inserta la información en la base de datos.

4. Cuando un usuario entra en TFGdashboard hará distintas consultas a la base de datos para mostrar la información (datos recolectados, cuentas de usuario, configuración, etc).
5. El paso final será poder visualizar los datos recolectados para poder estudiarlos.

La mayoría de los servicios se proveen desde un servidor como:

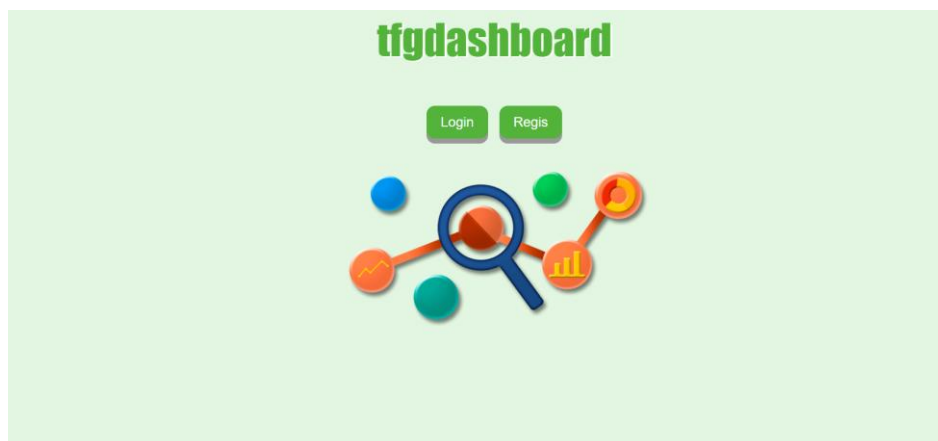
- El acceso a TFGdashboard.
- La página Ecommerce de muestra.
- La plantilla del core avcs.
- El alojamiento de la base de datos.
- Peticiones a la base de datos.

3.4.2 Diseño web

El contenido de la página de TFGdashboard está desarrollado principalmente en php y html, sin embargo, se sirve de otras herramientas y librerías como JavaScript, jQuery, Bootstrap y Firebase[11].

A continuación, se mostrará cómo se diseñó la plataforma TFGdashboard explorando cada una de sus pantallas y sus funcionalidades:

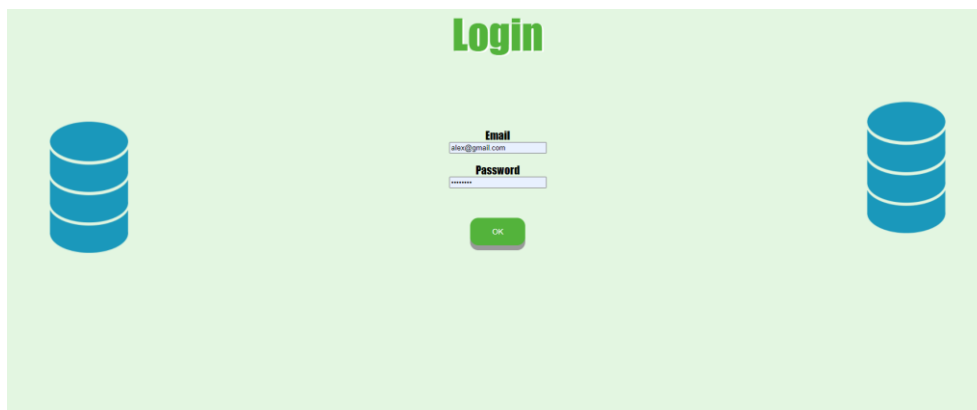
Inicio



3.6.3.1 TFGdashboard - Inicio

Es la página inicial de la desde donde se puede loguear o registrar.

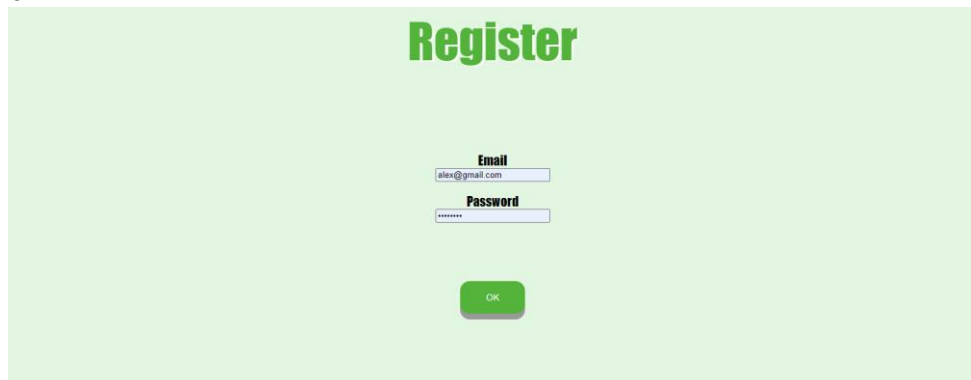
Login



3.6.3.2 TFGdashboard - Login

Desde la página de login un usuario puede introducir sus credenciales para acceder a las funcionalidades y servicios de TFGdashboard.

Registro



3.6.3.3 TFGdashboard - Registro

Desde la página de registro un usuario puede darse de alta en la plataforma con su email y una contraseña. El registro utiliza tecnología Firebase [12].

Home



3.6.3.4 TFGdashboard - Home

En la página de home el usuario tiene a su disposición acceder a todas las funcionalidades y servicios que ofrece la herramienta. El mapa que aparece es un iframe de código abierto que muestra cuantos usuarios acceden a TFGdashboard y su procedencia.

Administration



3.6.3.5 TFGdashboard - Administration

Desde esta página los usuarios tienen la posibilidad de crear y eliminar vistas o propiedades hasta alcanzar el límite que se sitúa en 5 propiedades con hasta 5 vistas cada uno.

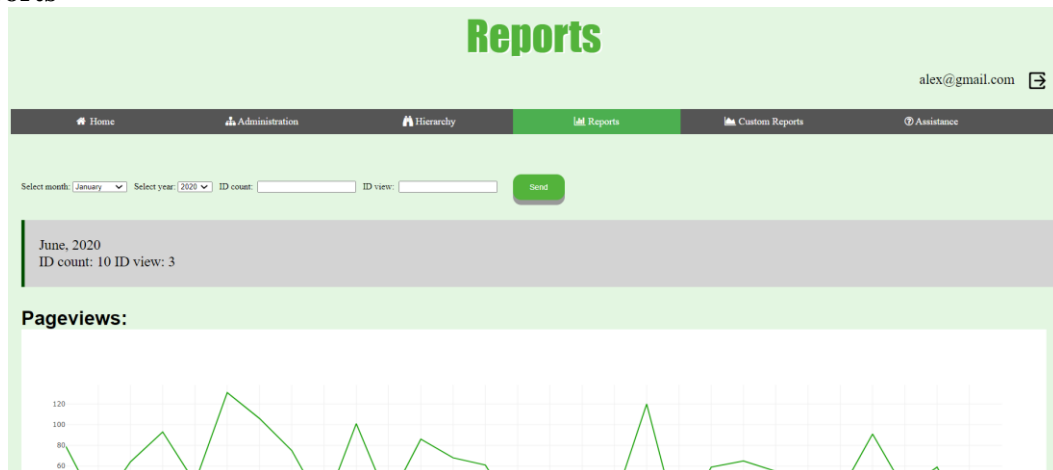
Views

Views				
alex@gmail.com				
Home	Administration	Hierarchy	Reports	Custom Reports
Assistance				
tfctest3 /id: 9	tfctest1 /id: 10	tfctest5 /id: 12	tfctest6 /id: 13	test9 /id: 58
-	produccionId 2	produccionId 7	-	-
-	preproduccionId 3	preproduccionId 8	-	-
-	testId 4	cuentaNuevaId 13	-	-
-	extemoId 5	-	-	-
-	intemoId 6	-	-	-

3.6.3.6 TFGdashboard - Views

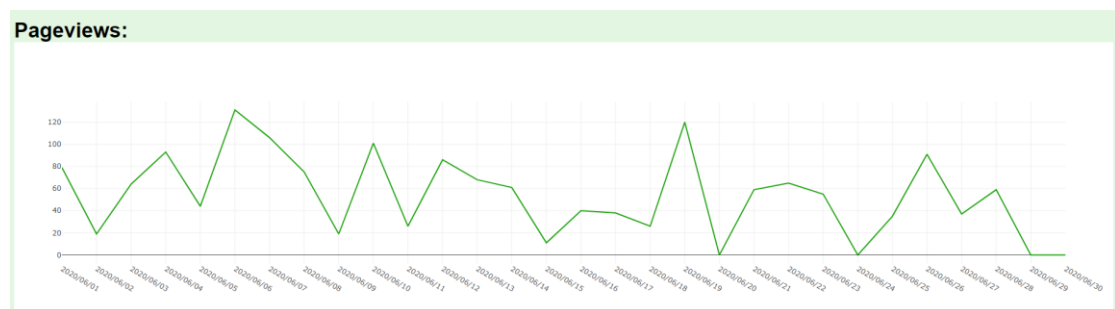
Se muestra en una tabla la distribución e información de las vistas y cuentas asociadas al usuario.

Reports



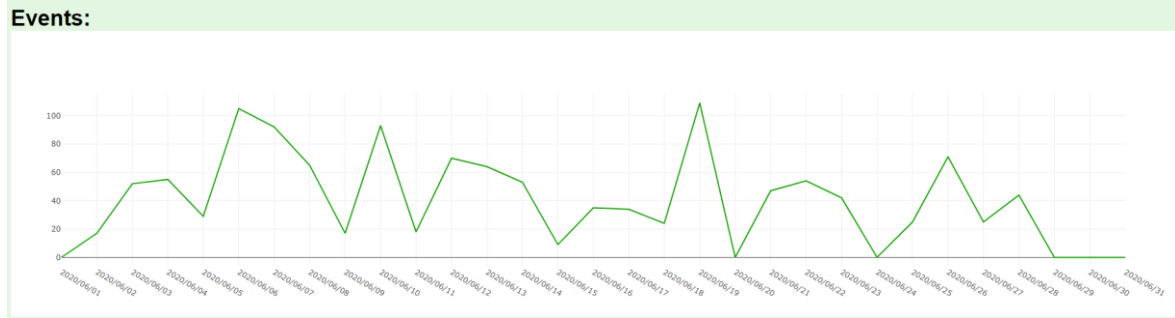
3.6.3.7 TFGdashboard - Reports

Desde esta página el usuario podrá acceder a una serie de informes predeterminados que ofrece la herramienta solo indicando el mes, el año y los id de la cuenta y la vista. Los gráficos proporcionados son:



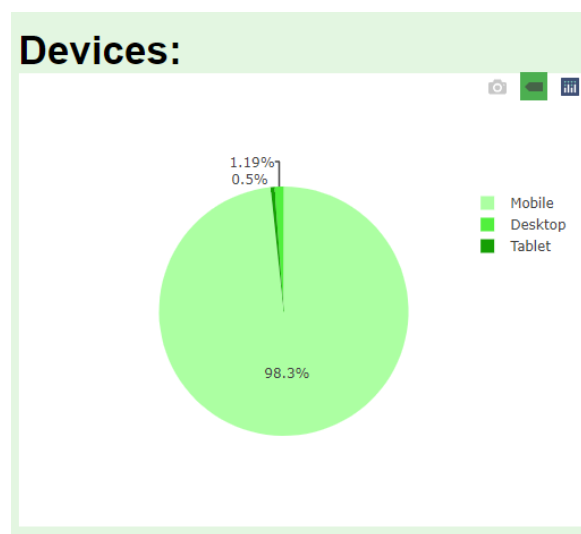
3.6.3.8 TFGdashboard - Pageviews

Muestra el total de páginas vistas que se han producido a lo largo de un mes.



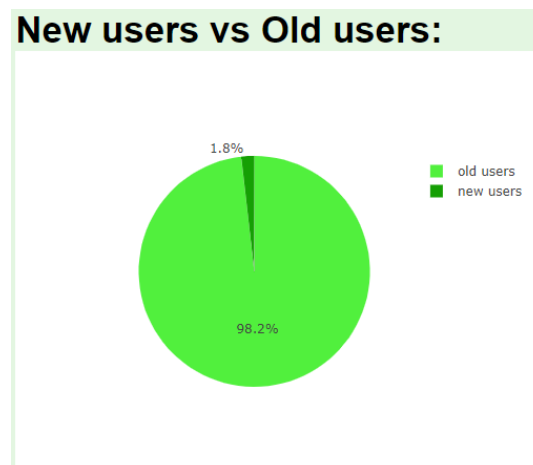
3.6.3.9 TFGdashboard - Events

Muestra el total de eventos producidos a lo largo de un mes sin sincluir las páginas vistas.



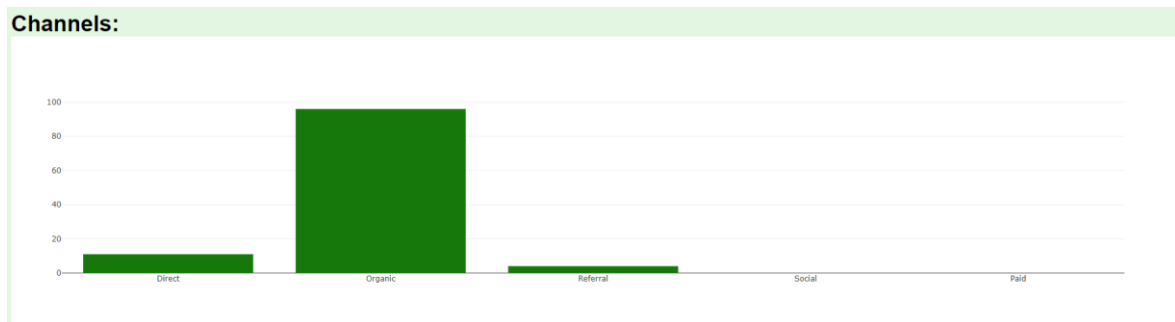
3.6.3.10 TFGdashboard - Dispositivos

Recupera la información de los dispositivos através de los cuales se han conectado los usuarios.



3.6.3.11 TFGdashboard - Usuarios

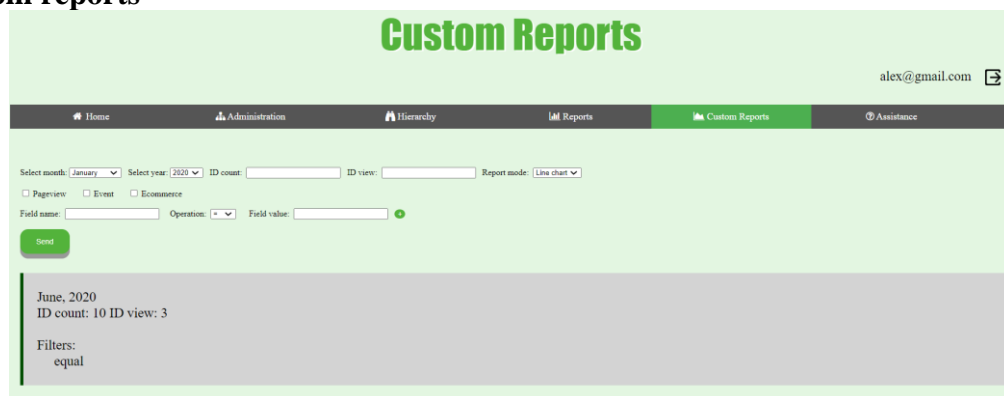
Muestra el porcentaje de los usuarios nuevos frente a los usuarios recurrentes que han accedido a la página, para poder tener esta información es necesario activar el tag con la cookie de nuevos usuarios (4.2.2 Cookie y 4.2.3 Nuevos usuarios).



3.6.3.12 TFGdashboard - Canales

Muestra los medios através de los que los usuarios han accedido a la página web.

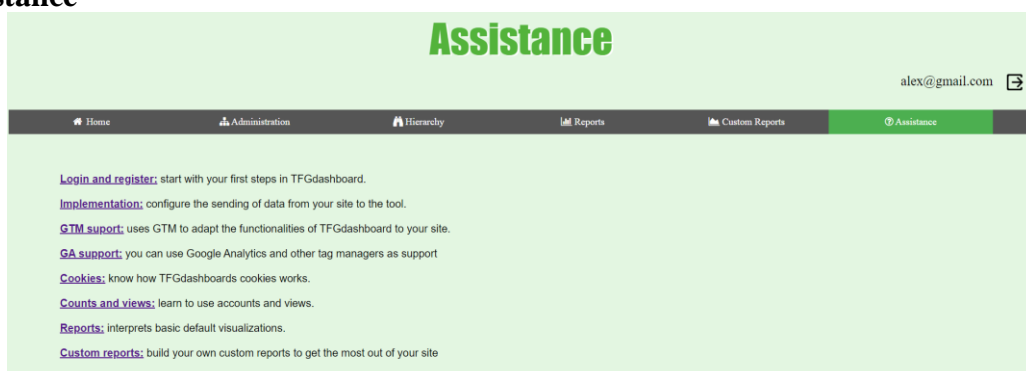
Custom reports



3.6.3.13 TFGdashboard - Informes

Esta pestaña permite a los usuarios crear sus propios informes dependiendo de la información que quieran ver y de como la quieran ver. Ofrece diferentes modos de visualización como el gráfico lineal, el gráfico de barras o en foema de tabla. También permite utilizar filtros para acotar las búsquedas y resultados para así poder hacer distintos estudios de los resultados.

Assistance



3.6.3.14 TFGdashboard - Ayuda

Ofrece toda la documentación relativa al uso y funcionamiento de la herramienta.

3.4.3 Identificación

Para llevar a cabo el registro y el login de los usuarios en la herramienta utilicé la tecnología facilitada por el módulo de Firebase ya que es un servicio gratuito y bastante potente ofrecido por Google, y que además proporciona servicios para la autenticación de usuarios mediante contraseñas, números de teléfono y otros proveedores de identidad (Google, Facebook o Twitter), pero en este caso se ha optado por utilizar un correo y una contraseña.

Firebase posee una consola para facilitar la gestión de los usuarios siendo posible inhabilitar o eliminar cuentas creadas y ver su información.

Además Firebase provee más funcionalidades que podrían implementarse en el futuro como poder logarse y registrarse con otros proveedores, o establecer una confirmación mediante email del registro.

Por último destacar que este módulo permite descargar de espacio a la base de datos ya que se ahorra el tener que almacenar también los datos de usuario.

3.4.4 Base de datos

Para almacenar todos los datos se creó una base de datos MySQL [13] porque ya tenía experiencia conectando desde php y creo que su rendimiento podía ser óptimo para este proyecto dada su sencillez y la cantidad y calidad que hay acerca de MySQL debido a que es gratuita y código abierto.

3.4.5 Gestión

De cara a que los usuarios puedan mantener un orden y puedan acceder a los contenidos con la mayor sencillez y eficiencia posible se ha diseñado este esquema de jerarquía (Anexo C) basado en los que utilizan otras herramientas similares como Google Analytics y Adobe Analytics.

User

El user es la cuenta individual de cada usuario de TFGdashboard. Cada usuario tiene una dirección de correo y una contraseña correspondiente.

Un único usuario puede tener hasta 5 cuentas.

Count

Es una cuenta donde engloba un activo determinado y donde recogerán todas sus trazas.

Una cuenta puede tener hasta 5 vistas.

View

Funcionaría como un primer filtro dentro de una cuenta ya que se podría elegir que solo llegarán determinadas trazas de un activo. Por ejemplo, se podría tener una vista solo para las trazas del entorno de producción y otras para el entorno de desarrollo. Esto se podría configurar desde la implementación de la función `_sendTo()` (3.6.10 Implementación):

Una cuenta puede tener hasta 5 vistas.

3.4.6 Gráficos

Para representar los datos se ha optado por la utilización de los gráficos lineales, de barras y circulares proporcionados por la librería Plotly.js, y tablas construidas mediante HTML y php.

3.4.7 Canales

Los canales [15] son una parte fundamental de la analítica ya que permite saber cuál fue el origen del usuario, es decir, a través de que medio el usuario accedió a la página web. Para poder tener este dato hay que implementar un tag específico en Google Tag Manager (se indicará en la parte de desarrollo) pero lo que hace es implementar una función que determine el canal del usuario y envíe un hit con la información del canal la primera vez que el usuario acceda a la web. Se hace solo la primera vez que acceda a la web para evitar el envío recursivo de hits que además serían erróneos.

4 Desarrollo

4.1 Documento de recolección

Basándome en la estructura de variables definida he construido el documento de recolección para tener una guía de las variables recogidas en cada página de la web sobre la que se está trabajando.

La finalidad de este documento es definir para todas las páginas vistas y eventos que variables van a llevar asociadas, que valor van a tomar y dar algo de información acerca de cada uno de ellos, ya que este mismo documento puede ser la referencia para los desarrolladores que implementen la solución.

Se incluyen varias hojas en el documento:

- **General:** contiene toda la información de la implementación de la analítica de la página.
- **Variables:** explica el significado y contenido de cada variable.
- **Leyenda:** proporciona una leyenda de colores para unificar.

Se incluyen en los anexos C, D, E y F un ejemplo del documento de recolección realizado sobre la web de prueba para tener una visión total del resultado. Sin embargo, a continuación, se explica en detalle todos los valores de la pestaña **General** de este documento.

4.1.1 Índice

Esta parte se encuentra anclada en la parte izquierda del documento para permitir avanzar por las páginas y eventos y poder seguir su referencia.

Su objetivo es mostrar todas las variables definidas encabezadas por el nombre de su grupo (content, ecommerce, custom, ...) de forma que todas las variables que quedan por debajo se engloban dentro de él.

En cuanto a las columnas, la primera muestra el nombre de la variable, la segunda el índice con el que se definirá la custom dimensión en Google Analytics y la tercera la categoría en la que se engloba dentro en TFGdashboards que pueden ser en pageviews, eventos o ecommerce:

- **Pageview (P):** hits enviados informando los datos de la carga de página (Anexo C).
- **Eventos (EV):** información de acciones que ocurren en la página (Anexo D).
- **Ecommerce (EC):** indica datos relacionados con acciones de comercio electrónico (Anexo E)

4.1.2 Encabezado

El encabezado mostrará información descriptiva sobre cada página y evento de la página:

Hit				Página	Página
URL				https://tfqdashboard.de/ecommerceTest/menu.php	https://tfqdashboard.de/ecommerceTest/prod1.php
Descripción				Página de inicio	Detalle de producto
Impl GA					
Impl AVCS					

3.2.2.1 Documento de recolección – Encabezado

En este ejemplo se muestra el encabezado de dos páginas. La parte izquierda es estática e indica los parametros que en la parte de la derecha se irán rellenando por columnas. A continuación se detalla el significado de cada campo:

- **Hit:** indica si es una página vista o un evento.
- **URL:** proporciona una dirección de la página para tener una referencia.
- **Descripción:** debe proporcionar algunos detalles o indicar que representa la página o evento.
- **Impl GA:** indica el estado de la implementación de esa página o evento en GA. Más abajo se indicará la convención de colores.
- **Impl AVCS:** indica el estado de la implementación de esa página o evento en TFGdashboard. Más abajo se indicará la convención de colores.
-

En lo relativo a la convención de los colores que se utilizan para indicar el estado de una página o evento son los siguientes:

- **Blanco:** es el estado inicial en el cual la página o evento se ha definido, pero no está aprobado para implementar.
- **Azul:** el evento o página está aprobado, pero todavía no se ha implementado.
- **Verde:** el evento o página se ha implementado y funciona de forma correcta.
- **Naranja:** el evento o página está implementado, pero está pendiente de revisión ya sea porque no se ha comprobado su funcionamiento o porque se ha observado un comportamiento inusual con el tiempo.
- **Rojo:** el evento o página está implementado, pero no funciona correctamente.

General	Sin revisar	sin implementar	Ok	Revisar	Mal
---------	-------------	-----------------	----	---------	-----

3.2.2.2 3.2.1.1 Documento de recolección – Leyenda

Además, también se incluyen al final columnas indicando como se deben informar las variables para el enhanced ecommerce de Google Analytics (Anexo F).

4.2 Web de muestra

Para el desarrollo de la web de prueba se ha utilizado una web con implementaciones muy básicas, pero con todas las funcionalidades básicas para la implementación de una medición e intentando recrear las condiciones lo más precisas posibles. Solo faltarían implementar funcionalidades relacionadas con el módulo de Google Analytics como son las eliminaciones de productos de un carrito que no se añadieron por falta de tiempo, y la medición de clicks e impresiones de promociones, que en una web de prueba son más complicadas de diseñar ya que suelen requerir de otras herramientas de gestión de campañas como Adwords [16] o Google Campaign[17].

En cualquier caso, es muy habitual que las páginas incluyan un objeto JavaScript añadido por los desarrolladores que contiene todos o casi todos los datos para la medición de la página para después poder utilizar estos datos para informar los datos a las herramientas de analítica. Esto se debe a que muchos datos provienen de fuentes externas a la web como base de datos con la información de productos o los datos de un usuario registrado, entonces son los desarrolladores lo que implementan las llamadas a estos servicios y cuando los tienen construyen el objeto para que se pueda utilizar y recoger los valores.

En este caso, como uno de los objetivos es mostrar como una web que no está preparada para implementar una medición puede utilizar el core avcs para llevar a cabo esta tarea se

ha utilizado un objeto lo más sencillo posible e informando solo lo imprescindible al que se puede acceder a través del objeto JavaScript **window._info**.

Esta es una práctica muy habitual ya que los valores que recupera el core avcs son los que aparecen en el DOM, y por ejemplo el id de los productos no es algo que suele aparecer, pero las páginas siempre lo tienen que acceder a él para poder mostrar la información del usuario. Por eso se suele utilizar este objeto para recuperar esta información.

También destacar que este escenario que se ha planteado es en el que menos información hay para recuperar los valores de la página, pero en otros casos este objeto contiene una gran cantidad de información, y cuanto más mejor ya que siempre que la información sea suministrada por los desarrolladores es más seguro que recogerla utilizando jQuery o JavaScript.

En este caso también se podría utilizar el core avcs sin ningún problema, solo sería necesario asignar los valores a los campos del core avcs desde este objeto suministrado por los desarrolladores.

4.3 SDK (core avcs)

Se ha creado el SDK core avcs cuya principal funcionalidad es construir un objeto JavaScript llamado **avcs** que se utilizará para centralizar al máximo todas las funcionalidades necesarias para la analítica de la página, desde la recolección de datos hasta enlazar con los tag managers. Además, se ha optado por utilizar una estructura de objeto JSON para mejorar los accesos a todos los niveles del objeto.

Este objeto se integra en la página a través de una etiqueta de Google Tag Manager y lo ideal es hacerlo lo antes posible en la carga de página para poder acceder a los datos recolectados lo antes posible y de esta forma no perder información debido a los tiempos de carga de la página.

Cada uno de los módulos cumple con una funcionalidad diferente que se explican en detalle a continuación.

4.3.1 Config

Engloba variables de la configuración del SDK con información como la versión actual del SDK, si está activado el ecommerce o estilos para facilitar las funciones de debug.

Pero hay dos variables que hay que destacar por encima de las demás:

Tms: esta variable contiene todas las herramientas de analítica soportadas por el SDK para enviar datos (TFGDashboard, Google Analytics, Tealium, Adobe Analytics y Ensign) aunque actualmente solo están desarrolladas TFGDashboard y Google Analytics ya que son las únicas accesibles desde el mercado. Para activar cualquiera de los envíos solo hay que ponerlos a true desde el código (avcs.config.tms).

Ecommerce: esta funcionalidad todavía no está implementada, pero se encuentra en desarrollo. Su función sería la de activar el envío del enhanced ecommerce para Google Analytics si está a true. Esto se conseguirá cuando se haya implementado un módulo en el SDK para el manejo del enhanced ecommerce, que es una de las mejoras pendientes.

4.3.2 Ecommerce

Esta función está destinada a manejar todas las configuraciones y acciones del enhanced ecommerce de Google Analytics. La idea sería que acepte como parámetros la acción, los productos asociados y otras variables de configuración, que de estos tres parámetros solo sería necesario indicar manualmente la acción ya que los demás se pueden extraer del propio core avcs.

Finalmente, dentro de la función dependiendo de la acción se formaría la acción correspondiente y se asignarían las variables para hacer el envío a Google Analytics. Está es una de las funcionalidades pendientes de implementar ya que de momento se está haciendo en un tag independiente dentro de Google Tag Manager.

4.3.3 Helpers

Este módulo contiene una serie de funciones que pretenden facilitar tareas habituales relacionadas con la recolección o tratamiento de datos y que se aplican principalmente sobre las variables del objeto **avcs** para la obtención de datos o para modificarlos. Las funciones contenidas son las siguientes:

beautifierObj

Recibe un objeto JavaScript y lo recorre tratando los datos para que dejarlos en la forma óptima para su manejo y envío a la herramienta de analítica correspondiente. Este tratamiento se aplica a las variables de **avcs** y consiste en la eliminación de mayúsculas y caracteres especiales para poder agrupar mejor los datos en la herramienta de analítica y no tener discrepancias.

beautifierString

Recibe una cadena de caracteres y devuelve la misma pero en minúsculas y eliminando tildes y caracteres especiales que se puedan detectar. Estos serían los remplazos que hace:

Equivalente	Carácter 1	Carácter 2	Carácter 3	Carácter 4	Carácter 5
a	ä	à	á	â	@
e	ë	è	é	ê	€
i	ï	ì	í	î	
o	ö	ò	ó	ô	
u	ü	ù	ú	û	

Esto es importante para unificar criterios a la hora de enviar los datos a las herramientas de analítica, ya que informar un mismo dato con mayúsculas o minúsculas será tomado como dos elementos distintos cuando en realidad son el mismo.

beautifier

Aplica las funciones de beautifier a las variables del objeto **avcs** del core **avcs**.

ceros

Recibe una cadena y un número que indicará la longitud de la cadena resultante que se desea. Con estos dos parámetros rellena la cadena recibida con tantos ceros como sea necesario hasta alcanzar la longitud indicada. Si la cadena ya tiene una longitud superior a la requerida no tendrá ningún efecto.

charToASCII

Recibe una cadena y la convierte a su valor ASCII.

detectBlocker

Se utiliza para detectar si el navegador del usuario tiene instalado una herramienta para el bloqueo de publicidad (AdBlocker). La idea es insertar un elemento vacío en la página y comprobar si es bloqueado, si es así es que hay un bloqueador funcionando.

getCurrentDate

Devuelve la fecha actual utilizando un formato YYYYMMDD (año, mes año).

Se ha elegido este formato para facilitar las comparaciones entre fechas ya que de esta forma se puede tratar la fecha completa como un número entero y hacer operaciones con ellos.

getGACookie

Obtiene el valor de la cookie de GA del cliente. Aunque sea una cookie de una herramienta como Google Analytics puede ser un dato muy útil para distinguir usuarios y en estudios más avanzados seguir el recorrido a través de una página en base a esta cookie.

getParam

Accede al parámetro de la URL que se le indica si existe. Se puede utilizar principalmente para acceder a los parámetros de utms que aparecen en las URLs asociadas a campañas.

getParameterByName

Devuelve el parámetro de la URL indicado utilizando una expresión regular.

getTime

Devuelve la hora actual en formato HH:MM:SS (horas, minutos, segundos).

getTipoDispositivo

Devuelve el tipo de dispositivo desde el que se está accediendo a la página dependiendo de si es un móvil, una tablet o un ordenador. Para esta clasificación se basa en el ancho de la pantalla siendo menor de 768px un móvil, entre 768px y 1054px una tablet y si es superior a 1054px un ordenador.

Este tipo de asignar el dispositivo a pesar de basarse un aspecto tan variable como el ancho de la pantalla es bastante acertado y solo comete errores con unos pocos de dispositivos.

getWeekday

Esta función indica el día de la semana que es en español. Se puede cambiar desde el propio código para que se muestre en otro idioma.

myCookie

Por defecto esta función viene desactivada, pero si se descomenta es perfectamente utilizable. Sin embargo, principalmente su función es generar un id único de usuario y si el usuario no tiene ya la cookie se le asigna la cookie con el id.

4.3.4 event

Esta variable se utiliza como clave para diferenciar entre tipos de eventos (pageview y generic) en la función avcs.push. En base a esta variable se hará llamar al tag manager con una clave de evento distinta para poder manejar el envío de datos como una página vista o como un evento genérico.

4.3.5 printOut

Esta función se utiliza para mostrar los mensajes del core avcs en la consola de desarrolladores y poder llevar un seguimiento de las acciones que se van ejecutando y poder verificar cuando puede haber un error.

Recibe 3 argumentos, el mensaje, el objeto avcs, y la categoría del mensaje que puede ser:

Error

Se utiliza para mostrar errores en el código o acciones que se han ejecutado de forma incorrecta.

```
avcs.printOut("mensaje de error", avcs, "error")  
Error: mensaje de error
```

3.3.2 SDK – printOut error

Warning

Mostrará mensajes si hay algún aspecto que no es correcto pero se puede seguir ejecutando.

```
avcs.printOut("mensaje de aviso", avcs, "warning")  
Warning: mensaje de aviso
```

3.3.2 SDK – printOut warning

Info

Su objetivo es mostrar mensajes meramente informativos para el seguimiento del código o valores de variables.

```
avcs.printOut("mensaje de informacion", avcs, "info")  
Info: mensaje de informacion
```

3.3.2 SDK – printOut information

Temp

Se utiliza para mostrar mensajes temporales durante el debuggeo de la web y no debería haber ninguno fuera del entorno de pruebas.

```
avcs.printOut("mensaje temporal", avcs, "temp")  
Temp: mensaje temporal
```

3.3.2 SDK – printOut temporal

4.3.6 init

Esta función es la primera que se ejecuta cuando se carga el avcs core en una página y se encarga de rellenar todas las variables disponibles llamando a la función setCustomVars, inicializa la variable evento como pageview e invoca a la función push para enviar el primer hit de pageview a las herramientas de analítica.

4.3.7 send

Esta función tiene como objetivo enviar los hits a la base de datos de TFGdashboards para poder visualizarlos en la plataforma.

Recibe como argumento las variables que se enviarán en el hit.

Al ejecutar la función **_sendTo()** deberá haberse incluido con anterioridad en la página dicha función, esto se explicará más adelante en la parte de implementación (3.6.10 Implementación).

4.3.8 push

La siguiente función se encarga de hacer las llamadas correspondientes dentro de cada herramienta de analítica activada en el core avcs (variable tms) para hacer los envíos de los hits.

Como primer paso tratará las variables y sus valores para eliminar mayúsculas y caracteres especiales, y después recorrerá las herramientas de analítica activas para proceder al envío del hit como corresponda.

Actualmente hay 2 herramientas activas:

- **Google Analytics**, utiliza `dataLayer.push()`.
- **TFGdashboard**, utiliza la función `_sendTo()`.

En ambos casos se diferencia entre dos tipos de hits, `pageviews` (hits con la información de la página que indica que se ha visualizado la página) y eventos genéricos (cualquier otra interacción en la página), que se diferencian por la variable evento que toma el valor del nombre del evento, pero para enviar los `pageviews` debe tomar el valor `'pageview'`. En cualquier caso, al final se reinicia esta variable para evitar propagarla erróneamente.

4.3.9 vars

Este objeto contiene todas las variables recolectadas en la página que serán inicializadas para su posterior envío a los tag manager. Las variables recogidas en la solución son las siguientes:

4.3.9.1 Content

Las variables de content son aquellas que hacen referencia a la propia página y su contexto, esto permite conocer el rendimiento de las páginas o grupos de páginas o categorizarlas para poder estudiar cómo se comportan los usuarios en determinadas páginas, por ejemplo, se podría conocer que tipología de páginas son las más visitadas para saber dónde podrían tener un mayor impacto los anuncios. Los valores que se recogen son los siguientes:

- **PageType**: clasifica las páginas por categorías para poder agruparlas por tipologías.
- **PageName**: construye una jerarquía para cada página basándose en la profundidad en la página y la tipología.
- **Hostname**: nombre del site.
- **URL**: dirección URL del site.
- **Referrer**: corresponde a la anterior URL visitada por el usuario.
- **Webmode**: indica el dispositivo desde el que se está a la web, puede ser `mobile`, `tablet` o `desktop`. Se basa en el tamaño de la pantalla del dispositivo así que no es completamente exacto, pero sí bastante fiable.
- **Language**: indica el lenguaje del navegador.
- **Category**: indica la tipología principal a la que pertenece la página actual.
- **Subcategory 1**: indica la primera subcategoría de la página.
- **Subcategory 2**: indica la segunda subcategoría de la página.
- **Subcategory 3**: indica la tercera subcategoría de la página. Es poco usual que se llegue a una profundidad de categorización tan alta.
- **Subcategory 4**: indica la cuarta subcategoría de la página. Muy pocas páginas llegan a esta profundidad, pero se añade siempre para tener margen.
- **Date**: fecha actual de la visita siguiendo el esquema `YYMMDD` (año, mes día).
- **Weekday**: marca el día de la semana. Preferiblemente en inglés.
- **Time**: indica la hora a la que se ha producido la visita a la página siguiendo el esquema `HH:MM:SS` (hora, minutos, segundos).
- **Adblock**: indica si la extensión `adblock` está activa en la página. Para ello intenta añadir un píxel en blanco a la página (que no influya en el funcionamiento de la misma) y si es bloqueado es que está activo.

4.3.9.2 User

Las variables de user hacen referencia a los valores propios del usuario actual de acuerdo a lo que se sabe de él. A un usuario no se le puede identificar con un nombre, pero si con id que podría estar conectado a otra información como si está suscripto a la newsletter o si tiene cuenta en la página. Normalmente para conocer estos valores hay que hacer uso de herramientas externas para que almacenen y después faciliten la información de los usuarios, pero es muy útil sobre todo para conocer qué tipo de promociones podrían interesarles o no. Los valores recogidos son los siguientes:

- **GA client:** almacena el valor de la cookie de GA. Este valor puede ser muy útil para identificar usuarios sin necesidad de que pasen por un proceso de login.
- **Useragent:** contiene información del usuario como su sistema operativo, software o el tipo de aplicación. También puede servir para identificar a los usuarios, aunque no de una forma muy precisa.
- **Newsletter:** indica si el usuario está suscrito a la newsletter.
- **Fingerprint:** constituye un id único que se asigna al usuario para mantenerle identificado sin necesidad de un login. No se llegó a implementar, pero la idea sería generar este id mediante una función hash con información como el useragent, el lenguaje o alguna cookie.
- **Idcrm:** id que proviene de un CRM en caso de estar implementado (es complicado encontrarlo implementado ya que es un aspecto bastante avanzado).
- **Cookie:** este espacio está reservado para guardar el valor de una cookie a elección de la implementación.

4.3.9.3 Traffic

Las variables de traffic tienen como objetivo identificar el tipo de tráfico que hay en la web y su procedencia. Su utilización se restringe a campañas y promociones ya que permite conocer si el usuario ha accedido a la web a través de un enlace por el cual se ha pagado o una red social, por ejemplo, y así conocer el impacto que tienen las campañas sobre el tráfico que llega a la web. Las variables que toman valor son las siguientes:

- **Campaign:** identifica la campaña o promoción a través de la cual se accedió a la página.
- **Source:** identifica la fuente del tráfico como el buscador o la newsletter.
- **Médium:** identifica el medio a través el cual se accedió a la página.
- **Term:** indica el término de la promoción. Se utiliza para las búsquedas monetizadas.
- **Content:** se utiliza para diferenciar anuncios o links de cara a hacer test AB. Estos test consisten en abrir dos o más versiones de una misma página con algunas diferencias y estudiar cuales tienen mejor rendimiento.

4.3.9.4 Search

Estas variables se corresponden las búsquedas internas del site y los resultados:

- **Keyword:** palabra o palabras por la que se realizó la búsqueda.
- **Resultnumber:** número de resultados obtenidos en una búsqueda.
- **Originpage:** página desde la que se realizó la búsqueda, aquí queda a elección de la implementación si utilizan el pageName, pageType o la URL.

4.3.9.5 Data quality

Las siguientes variables son meramente informativas para los desarrolladores, tienen muy poco valor de cara al estudio de la página. Se utilizan especialmente para la detección de errores para conocer la versión implementada:

- **Gtmversion:** indica la versión de GTM o del tag manager donde está implementado.
- **Tfgversion:** indica la versión del core avcs que se está utilizando.

4.3.9.6 Products

Pertenecen a las variables propias de productos. Para seleccionarlas he utilizado el modelo propuesto por Google para el enhanced ecommerce que propone una serie de variables acotadas y estáticas pero que se informarán de forma exclusiva dependiendo la acción realizada por el usuario. Este módulo permite medir visualización de productos en páginas de detalle o listados, y clicks en productos, y aprovechando que Google ofrece este modelo para hacer la medición creo que lo más efectivo es seguirlo para facilitar los datos lo más directamente posible:

- **Name:** nombre comercial del producto.
- **Id:** identificador único para ese producto.
- **Price:** precio individual del producto.
- **Brand:** modelo o marca del producto.
- **Category:** categorización del producto dentro del modelo.
- **Variant:** variante del producto, puede ser un color u otra seña identificativa que no se recoja por otra variable.
- **List position:** en caso de que se encuentre en un listado con otros productos se indicaría su posición en la lista.

4.3.9.7 Promotions

Las siguientes variables indican los valores que se les asignan a las promociones internas de la página que pueden estar en forma de banner o enlace externo. Para seleccionarlas he utilizado el modelo propuesto por Google para el enhanced ecommerce que facilita un conjunto de variables cerradas para la medición de visualización y el click de promociones:

- **Id:** id que identifica la promoción.
- **Name:** nombre comercial que se le da a la promoción.
- **Creative:** nombre de la creatividad, se utiliza para categorizar las promociones.
- **Position:** indica con un índice la posición de la promoción dentro de la página como si se tratara de un listado.

4.3.9.8 Ecommerce

Google también tiene un modelo de variables en el enhanced ecommerce para informar acciones relacionadas con las transacciones como los pasos de compra de una transacción o la confirmación cuando una transacción se realiza con éxito. Por lo tanto se utilizará el mismo modelo propuesto con el fin de poder utilizarlo para enviar los datos a Google Analytics de manera directa:

- **Transactionid:** indica el identificador único de la transacción.
- **Price:** precio de la transacción excluyendo impuestos o descuentos.
- **Step:** número del paso en el proceso de pago.
- **Quantity:** cantidad de productos en la transacción.
- **Shipping:** precio del envío de la orden.

- **Discountvalue:** valor del descuento que se aplica a la transacción si lo hubiera.
- **Discouncode:** código de descuento que se aplica en la orden si lo hubiera.
- **Tax:** impuestos que se aplican a la orden.

4.3.9.9 Custom

En este apartado se añadirían todas las variables que no se contemplarán anteriormente pero que fueran imprescindibles para la medición del site. Si la medición necesitará de demasiadas variables custom podría plantearse añadir un nuevo módulo a los anterior con estas variables e incluirlas en el SDK más adelante, pero en todo caso sería algo propia de dicha implementación e independiente del SDK general.

4.3.9.10 Events

Las siguientes variables se utilizan para categorizar los eventos dentro de GTM, y aunque se utilizará otro tag manager es muy recomendable implementarlas para tener una diferenciación clara de los eventos y poder hacer un estudio de los mismos:

- **EventName(evento):** es la clave del evento que diferencia los eventos de las páginas vistas en GTM. Se recomienda usar las menos posibles para tener una solución en el tag manager lo más general posible y por lo tanto mejorar su mantenibilidad.
- **EventCategory:** indica la categoría del evento como pueden ser eventos de compra, acciones de usuario o acciones de configuración, pero pueden ser muchos otros.
- **EventAction:** sería el siguiente nivel de categorización ya que indica la acción que se ha realizado.
- **EventLabel:** este término se utiliza para diferenciar las dos anteriores variables ya que muchas veces pueden coincidir, pero se quiere hacer una última categorización.

4.3.10 setCustomVars

Esta función está reservada para hacer la asignación de los valores iniciales en página de las variables de avcs.vars. Aquí se recomienda utilizar herramientas como **jQuery** para acceder a los elementos del DOM y hacer uso de las funciones del apartado de **helpers** para facilitar la recolección de datos.

4.4 Google Tag Manager

4.4.1 Seguimiento

Para la implementación de Google Tag Manager me apoyé en la documentación oficial que ofrece Google donde se detallan los pasos a seguir.

Principalmente elegí el tipo de contenido correspondiente a mi web y añadí el código de seguimiento en <head> y <body> de cada página (Anexo M).

4.4.2 Cookie

Dentro de Google Tag Manager se han implementado una serie de tags que son completamente opcionales pero que ayudan a TFGdashboard a hacer un seguimiento más completo. Se consideran opcionales porque se basan en su mayoría en el uso de cookies en la propia página, lo que hace que si se implementa en una página real fuera necesario tener un gestor de cookies que advirtiera del uso de cookies y permitiera su gestión, y esto no es algo que todas las webs poseen de momento por lo que se ha preferido excluir de la implementación principal.

Las cookies [18] y storages [19] implementados son los siguientes:

`_tfgSession`

Pertenece a los `sessionStorage` por lo que expira con la sesión. Únicamente sirve para comprobar si cuando carga una página la sesión es nueva (ya existe el storage) o no.

`tfgCookie`

Pertenece a los `sessionStorage` y no tiene una expiración. Contiene un identificador único (un número aleatorio de 16 cifras) y se utiliza para identificar a los usuarios, se envía en los hits a la herramienta TFGdashboard.

`cart`

Pertenece a los `sessionStorage` y no tiene una expiración, pero se borra cuando se llega a la página de confirmación de compra. Únicamente almacena el contenido del carrito del usuario y no se informa a ninguna otra herramienta.

`_GA`

Esta cookie es propia de la herramienta de Google Analytics y no conveniente modificarla ya que la maneja la propia herramienta y la utiliza para generar sus informes. En este proyecto finalmente no se llegó a utilizar, pero podría haberse utilizado para identificar usuarios o para generar un fingerprint junto con otros valores.

4.4.3 Nuevos usuarios

Un dato muy útil sobre una página web es cuantos usuarios de los que llegan a la página son nuevos y cuales recurrentes, ya que tener muchos usuarios nuevos indican que la web se está expandiendo a nuevas audiencias o puede servir como medidor del alcance que está teniendo una campaña, mientras que tener muchos usuarios recurrentes indica que la web ha conseguido establecer un vínculo con los usuarios y hace que estos quieran volver, se les está ofreciendo algo que les gusta.

Por estas razones me pareció importante tener estos datos en TFGdashboard y es algo que no se podía medir con eventos comunes, si no que llevaba hacer una implementación a parte de la recolección de datos y los eventos del documento de recolección, por ello se implementó en un tag individual en Google Tag Manager.

La idea es comprobar si el usuario tiene la cookie **`tfgCookie`**, si no quiere decir que es la primera vez que el usuario accede a la página, por lo que se enviará un evento indicando que el usuario es nuevo (`newUser`) y el canal por el que accedió a la página.

Si ya posee la cookie, entonces se indicará que el usuario recurrente con un evento similar (`oldUser`).

Pero hay que evitar que estos eventos se lancen con cada carga de página ya que al usuario solo hay detectarlo en la primera página, el resto son irrelevantes. Por ello se utiliza un storage de sesión que se implementa siempre que no exista, por lo que siempre primero se comprueba si existe, si es así es que no es la primera página del usuario y no hay que hacer nada, si no existe se actúa como se mencionó antes comprobando si es un usuario nuevo o recurrente.

4.4.4 Implementación

Para integrar los envíos de los hits a Google Analytics se utilizó la herramienta Google Tag Manager donde se integraron algunos scripts como el core avcs y se activaron ciertas configuraciones que se abordarán a continuación:

4.4.4.1 Core avcs

Se descargó el código del core avcs (<https://tfgdashboard.de/tfgdashboard/coreAvcs.js>) y se adaptó a la página web de prueba personalizando ciertos aspectos como la recolección de variables y activando los tag managers que se van a utilizar.

Es importante elegir cuando se carga el core avcs ya que debe hacerse lo antes posible, pero deben estar disponibles las variables, en todo caso la carga del core avcs marcará los tiempos del resto de scripts y hay 3 posibles momentos:

- **All pages:** se cargará en cuanto cargue el contenedor de Google Tag Manager, es lo más pronto posible.
- **Dom ready:** se cargará cuando se haya cargado por completo el DOM de la página, sería un momento intermedio.
- **Window loaded:** se cargará cuando se cargue la página al completo y no se recomienda cargar más tarde de este momento.

En este caso sea optado por utilizar el DOM ready ya que utiliza jQuery para acceder a ciertas variables y si queremos recogerlas es necesario que este cargado todo el DOM.

Lo ideal sería poder cargarlo en **All pages** ya que puede darse que un usuario entre en la página y salga antes de que haya cargado el DOM por distintas razones y eso haría que se perdieran datos de pageviews. Para que se diera este caso sería necesario que desde la propia página se facilitaran las variables en un objeto JavaScript antes de la llamada a Google Tag Manager para tener las variables ya disponibles en cuanto carga.

En el caso de esta web no se facilitan todas las variables para mostrar cómo se podría hacer la recolección de datos desde el modelo más básico.

4.4.4.2 Variables

Como primer paso se ha creado una variable en Google Tag Manager por cada variable [3] recogida en el core avcs.

4.4.4.3 Pageviews

Con el core avcs implementado ya se pueden comenzar a medir las pageviews de la página web, para ello se deben crear un tag, un trigger y una variable:

Variable

Lo primero de todo será crear una variable que contenga toda la información necesaria para los envíos de pageviews a Google Analytics, y eso incluye:

- Especificar la propiedad de Google Analytics
- Especificar el host
- Determinar los índices y los valores de las custom dimensions
- Activar las funciones de Enhanced Ecommerce

Trigger

Es importante crear un trigger [4] de activación para que se ejecute el tag del envío de pageviews. Para ello se le debe poner que se active cuando reciba como clave de evento **avcs.pageview**, ya que desde el core avcs se hizo el `dataLayer.push()` con ese valor en la clave evento cuando el hit es de página vista.

Tag

El tag [5] deberá ser de tipo Google Analytics: Universal Analytics que es el que permite hacer los envíos de datos a Google Analytics, se le asigna un tipo de seguimiento de Página vista (Pageview) y en la configuración de Analytics se le asocia la variable que se

creó anteriormente. Por último, como trigger de activación se utiliza el creado anteriormente.

4.4.4.4 Eventos genéricos

El siguiente paso sería habilitar en envío de hits de todos los eventos en página, para ellos es necesario crear un tag, un trigger y una variable de configuración para los eventos:

Variable

Igual que en el caso de las pageviews es necesario crear una variable con contenga la configuración de los hits de eventos genéricos. Pero en este caso he optado por algo más simple ya que no se van a informar ninguna de las variables personalizadas, si no que las únicas que se informarán serán el eventAction, eventCategory y eventLabel pero que se añadirán más adelante. También es importante habilitar aquí las funciones de enhanced ecommerce para poder obtener los beneficios de esta funcionalidad que es básica.

Trigger

Es importante tener un trigger para activar el tag y poder enviar los hits, y se utilizó la clave 'generic' que se estableció en el core avcs.

Tag

Para completar la configuración de los hits de eventos genéricos se añadió un tag del tipo Google Analytics: Universal Analytics con un tipo de seguimiento para Eventos, se incluyó la variable anterior con la configuración, y por último se asignó a la categoría, a la acción y a la etiqueta las variables correspondientes.

Al final considero que esta forma de implementar la medición para Google Tag Manager es la más beneficiosa, ya que requiere de solo dos tags, dos triggers y dos variables (las de configuración ya que hay que crear una variable por cada valor que se quiera enviar y esto es obligatorio) para medir todos los eventos de la página. Esto hace que sea un sistema muy mantenible y escalable porque con el tiempo Google Tag Manager tiende a complicarse por las distintas soliciones que se aplican, actualizaciones o simplemente que haya que hacer un traspaso a otro desarrollador, y de esta forma los elementos que hay son mínimos pero cumpliendo con todas las funcionalidades.

4.4.4.5 Ecommerce

El enhanced ecommerce de Google Tag Manager permite recoger datos e información relacionada con la compra y venta de productos, y visionado y clicks en promociones y en productos lo que permite tener una visión completa para poder sacar el máximo rendimiento de la página.

Es importante mencionar que se ha optado por implementar el enhanced ecommerce desde es eventos, es decir, todas las acciones se enviarán en hits de eventos sin utilizar las pageviews. Esto se ha hecho así por sencillez y será lo más utilizado, sin embargo, si fuera un sitio web con mucha actividad lo recomendable sería utilizar los pageviews para incluir todas las acciones posibles como las impresiones. Esto depende de la actividad del site porque Google pone un límite de hits por usuarios, sesiones y mensuales.

También cabe destacar que Google recomienda en su guía de implementación del enhanced ecommerce utilizar los pageviews siempre que sea posible, pero en sus muestras de como implementarlo utiliza solo eventos, así que ambas maneras son válidas y cada una se adapta mejor a distintos escenarios.

Para su implementación se ha utilizado solo un tag de tipo HTML personalizado donde se colocan los listeners con jQuery para las acciones. A continuación, se muestra la estructura que tienen las distintas acciones (Anexo L):

Impresiones

Las impresiones pueden ser de productos y promociones que se visualizan (las promociones no se implementaron aún) y tienen la siguiente estructura:

- **Currency code:** nombre de la moneda utilizada en la transacción (utiliza Euros)
- **Impressions:** contiene todos los objetos que se han visualizado.
 - **Name:** nombre del producto.
 - **Id:** identificador único del producto.
 - **Price:** precio unitario del producto.
 - **Brand:** marca del producto.
 - **Category:** categoría del producto.
 - **Variant:** variante del producto.
 - **List:** nombre de la lista donde se está visualizando.
 - **Position:** posición de la lista donde está apareciendo.

Páginas

Los eventos de enhanced ecommerce relacionados con las páginas son detalles de productos, los pasos del proceso de compra y la confirmación de las compras. Estos eventos se lanzan con la carga de página y se podrían incluir en los pageviews pero aquí aún se lanzan en eventos por sencillez.

Detalles de producto

Se activa en las páginas individuales de productos donde está toda su información en detalle.

Pasos del checkout

Se activa en cada paso del checkout sin contar la visualización inicial del carrito. La variable ‘step’ indica el paso del checkout donde se encuentra e irá variando.

Transacciones

Se activa en las páginas de confirmación de compras.

- **ActionField:** contiene la información global de la compra:
 - **Id:** identificador de la compra.
 - **Affiliation:** nombre de la empresa o página a la cual se le atribuye la venta.
 - **Revenue:** precio total de la compra incluyendo impuestos y gastos de envío.
 - **Tax:** impuestos asociados a la venta.
 - **Shipping:** gastos de envío.

Acciones

Estas acciones se sitúan normalmente en botones o banners de acción y se utiliza principalmente para medir productos añadidos y eliminados al carrito, y clicks en productos y promociones.

Añadir al carrito

Se activará el evento cuando se añada un producto al carrito de compra o cuando se aumente los productos en el carrito.

Eliminar del carrito

Se activará cuando se eliminen productos de carrito de compra o cuando se decrementen las unidades de un producto.

Clicks en productos

Se lanzará cuando desde un listado de productos se pulse en uno de los productos del listado para acceder a su información en detalle.

4.4.4.6 Listeners

Se ha utilizado un tag exclusivamente para incluir los eventos en la página, para lo que se ha utilizado jQuery [6] que permite implantar escuchadores en los elementos del DOM cuando se pulse o se pase sobre ellos, por ejemplo, pero hay muchas más posibilidades.

En definitiva, se aquí se colocan todos los escuchadores donde quiero que salten los eventos especificados en el documento de recolección.

El push se tiene que hacer sobre el objeto avcs facilitado por el core avcs, y como argumento es necesario indicarle la clave del evento, el eventCategory, eventAction y eventLabel para poder obtener el valor en las variables correspondientes.

4.5 Google Analytics

Para la visualización de los datos de esta web se ha creado solo una propiedad y una vista [8], siendo esta propiedad de tipo Web.

Esto se debe a que solo se pueden utilizar el tipo web o web+app [9], pero decidí utilizar el tipo web porque es un tipo de propiedad estable y totalmente funcional.

El tipo web+app también sería válido, de hecho es el tipo de propiedad que un futuro cercano está llamado a sustituir los otros dos, pero aún está en una versión de prueba, pero en próximos pasos sería un buen ejercicio implementar esta solución.

4.6 TFGdashboard

El objetivo de esta plataforma es poder visualizar los datos recolectados en página para poder hacer informes y estudiar los datos con el fin de mejorar el rendimiento del site.

4.6.1 Servidor

Para llevar a cabo el proyecto era necesario disponer de un servidor que permitiese alojar y proveer servicios a cualquier usuario como:

- El acceso a la plataforma TFGdashboard.
- La página Ecommerce de muestra.
- La plantilla del core avcs.
- El alojamiento de la base de datos.
- Peticiones a la base de datos.

Por ello contraté un servidor Hetzner [10] que debía cubrir las necesidades del proyecto, así que decidí contratar uno con las siguientes características:

- Un host web, tfgdashboard.de
- 100 GB de espacio en disco
- Posibilidad de alojar hasta 20 bases de datos
- Posibilidad de gestionar bases de datos MySQL y PostgreSQL
- Hasta 50 usuarios FTP
- Posibilidad de tener hasta 300 mailboxes
- Conexión SSH

El proyecto podría haberse completado con un servidor de características más básicas, pero no estaba seguro de cuál sería la magnitud del proyecto y preferí no arriesgar.

4.6.2 Base de datos

Finalmente se decidió la siguiente estructura (Anexo K):

User

Contiene información de los usuarios registrados en la herramienta.

- **id (PK)**: identificador único del usuario.
- **email**: email del usuario, será único.

Count

Cuentas creadas en la herramienta.

- **IdCount (PK)**: identificador único de la cuenta.
- **User**: identificador único del usuario.
- **Name**: nombre de la cuenta, será único **solo** en la misma cuenta.

View

Vistas creadas en la herramienta.

- **IdView (PK)**: identificador único de la vista.
- **IdCount**: identificador único de la cuenta.
- **Name**: nombre de la vista

TracePageview

Trazas de páginas vistas que han llegado a la herramienta.

- **IdTrace (PK)**: identificador único de la traza.
- **IdCount**: identificador único de la cuenta.
- **IdView**: identificador único de la vista.
- **Date**: fecha de generación de la traza (dd/mm/yy).
- **Hour**: hora de generación de la traza (hh:mm:ss) formato 24h.
- **UserID**: identificador del usuario según indica la cookie de GA.
- **EventName**: nombre del evento, en este caso será siempre Pageview.
- **Custom Fileds**: serán todos los valores definidos que se desean enviar a la herramienta.

TraceGeneric

Trazas de eventos genéricos que han llegado a la herramienta.

- **IdTrace (PK)**: identificador único de la traza.
- **IdView**: identificador único de la vista.
- **Date**: fecha de generación de la traza.
- **Hour**: hora de generación de la traza formato 24h.
- **UserID**: identificador del usuario según indica la cookie de GA.
- **EventName**: nombre del evento.
- **EventAction**: acción del evento.
- **EventCategory**: categoría del evento.
- **EventLabel**: etiqueta asociada al evento.

TraceEcommerce

Trazas de eventos de comercio que han llegado a la herramienta.

- **IdTrace (PK)**: identificador único de la traza.
- **IdView**: identificador único de la vista.

- **Date:** fecha de generación de la traza.
- **Hour:** hora de generación de la traza formato 24h.
- **UserID:** identificador del usuario según indica la cookie de GA.
- **EventName:** nombre del evento.
- **EventAction:** acción del evento.
- **EventCategory:** categoría del evento.
- **EventLabel:** etiqueta asociada al evento.
- **TransactionId:** identificador único de cada transacción.
- **Price:** precio total de la transacción.
- **Step:** paso del proceso de pago de una orden en número.
- **Quantity:** cantidad de productos asociados.
- **Shipping:** coste de envío de la transacción.
- **Discount value:** valor de descuento en una transacción.
- **Discount code:** código de descuento en una transacción.
- **Tax:** impuestos asociados a una transacción (IVA).
- **Product Id:** identificador único de un producto.

4.6.3 Gráficos

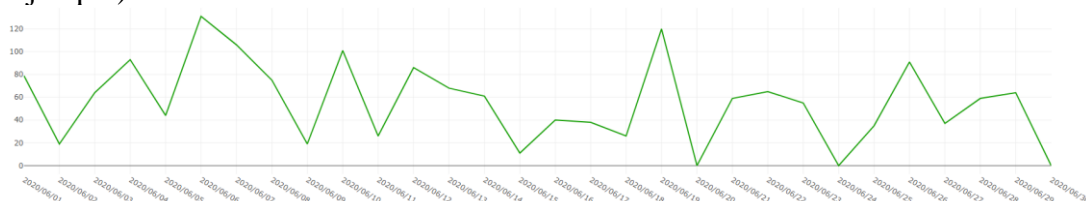
La visualización de gráficos se ha implementado para las pestañas de Reports y Custom reports con el fin poder mostrar los datos recogidos de una forma más amigable y facilitar el estudio de los mismos. Para ello se ha utilizado la librería JavaScript Plotly.js [14] que provee de diferentes tipos de gráficos (de barras, lineales o circulares).

De entre todas las opciones que ofrecía Plotly.js se han elegido tres con los que se pueden cubrir todos los tipos de estudios que desean realizar:

Gráfico lineal

Representa el acumulado diario de datos a lo largo de un mes y sirve para mostrar la progresión de los datos en el tiempo.

Permite ver caídas o incrementos bruscos para detectar posibles errores en los datos o simplemente tendencias (si un cambio en la web ha hecho crecer o disminuir las ventas, por ejemplo).



3.6.7.1 TraceEcommerce – Gráfico lineal

Gráfico de barras

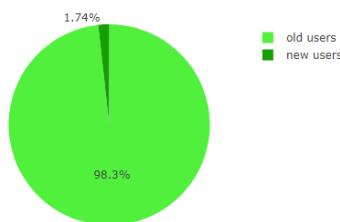
Muestra el acumulado de datos agrupado por categorías y es útil para mostrar rankings.



3.6.7.2 TraceEcommerce – Gráfico de barras

Gráfico circular

Muestra los valores de las categorías de un grupo de datos dentro de su totalidad. Es útil para ver como de presente esta una categoría dentro de un grupo y determinar cuál puede ser su peso.



3.6.7.1 TraceEcommerce – Gráfico circular

También hay que destacar que, aunque su implementación no se hizo con Plotly.js también existe la opción de representar los datos por tablas lo que permite entrar al detalle de los datos y cuál es su valor real.

Las **tablas** pueden ser el elemento más potente que hay en la herramienta para mostrar datos por el nivel de detalle al que llegan, pero no nos van a proporcionar los datos de forma global, si no individualizada.

Plotly.js también proporciona la siguiente barra de interacción en cada gráfico donde se puede descargar el gráfico, hacer zoo o moverse por el gráfico.

Por último, queda pendiente la extensión de estos gráficos a nuevos modelos que permitan hacer más estudios como los funnels y los gráficos en mapa.

4.6.4 Canales

Estas son las definiciones que se han utilizado para clasificar los canales:

Direct

El tráfico directo es aquel en que los usuarios introducen la dirección directamente en la URL y se ha definido como aquel en los que no existe una página previa a la entrada en la web. Aun así, puede tener el error de usuarios que tenga la web como página principal, aunque es muy poco común.

Organic

El tráfico orgánico es aquel que proviene de una búsqueda en un buscador como Google, Yahoo! o Bing. En este caso se han tomado los usuarios cuya página anterior a la entrada ha sido Google, Yahoo! o Bing, aunque quedan los posibles casos excepcionales como los usuarios que estando en una de estas páginas copian la dirección de la web en la URL.

Referral

El tráfico referenciado es aquel que proviene de enlaces en otras webs y no son redes sociales ni se está pagando por la navegación, como puede ser que a través de un acuerdo una web tenga un enlace a la mía a modo de colaboración, entonces estos usuarios entrarían por este canal.

Como posible error quedarían los usuarios que estando en otra página cualquiera peguen en la barra de direcciones la dirección de la web.

Social

El tráfico social es aquel que proviene de enlaces o banners en redes sociales, y aquí solo se están detectando aquellos usuarios que provengan de facebook, twitter, Instagram o linkedin por ser las redes sociales más conocidas.

Aquí también estaría el posible error de usuarios que estando en cualquiera de estas redes sociales peguen en la barra de direcciones la dirección de la web.

Paid

El tráfico de pago es aquel por el que se paga para colocar un banner o enlace en otras webs que lleven a la propia. En este caso se utiliza para detectarlo los parámetros de utm en la URL (utm campaign, utm source, utm term, utm content) que son los que se utilizan habitualmente para categorizar las campañas.

Los canales se han configurado siguiendo la definición de Google lo más precisamente posible, aunque siempre quedan posibilidades a que haya posibles errores al categorizarlos. Sin embargo, al estar implementados como un tag en Google Tag Manager estas reglas se pueden editar ya que no siempre se cumplen estas reglas completamente (todas las herramientas de analítica permiten editar la configuración de los canales), si no que a veces dependen de la naturaleza de la página web y el tipo de audiencia a la que va dirigida, por ejemplo el tráfico social puede considerarse de pago si se hace una campaña a través de redes sociales, o se pueden añadir más redes sociales a la lista.

Es importante destacar que para tener un seguimiento de los canales es necesario haber activado la propia de TFGdashboard que se explicará en próximos apartados (4.2.2 Cookie).

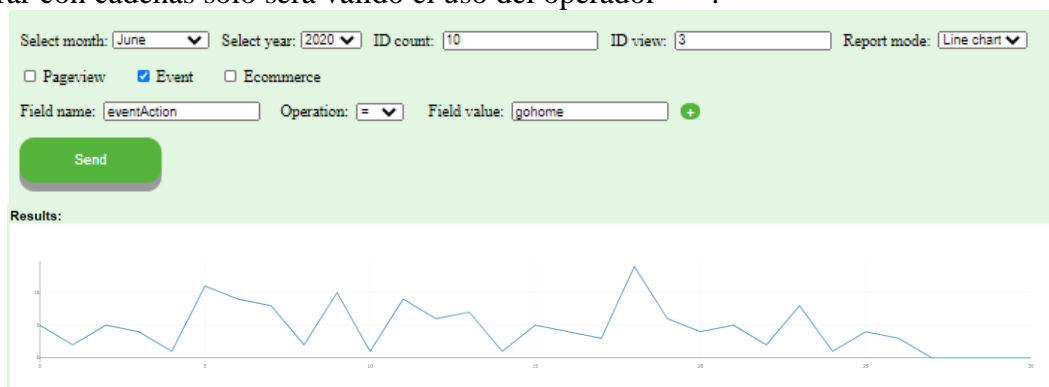
4.6.5 Filtros

Para crear los informes personalizados es importante utilizar los filtros ya que permiten acotar al máximo los resultados dependiendo de la finalidad de estos, ya sea para facilitar su visualización o para mostrar solo el grupo de datos que nos interesan.

El máximo de filtros aplicables es de tres (se pueden añadir filtros pulsando en el símbolo más al lado de los filtros) y para cada tipo de informe se ha establecido una configuración de filtros:

Gráfico lineal

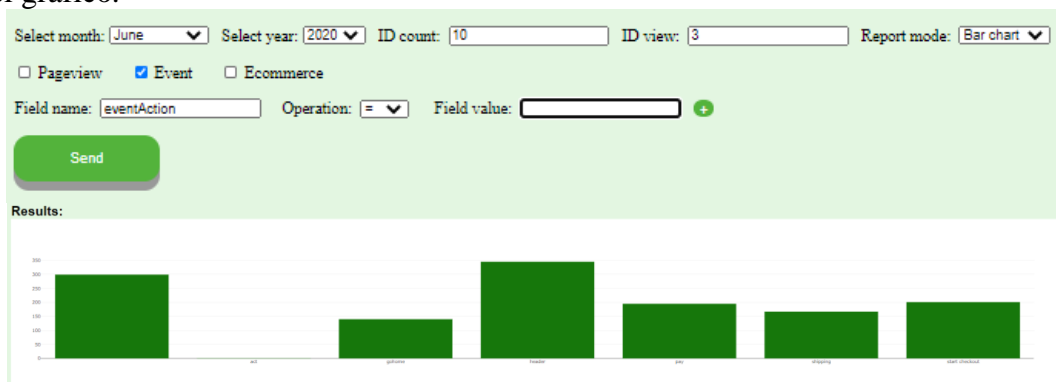
Introduciendo el campo y la condición que debe cumplir. Hay que tener en cuenta que para operar con cadenas solo será válido el uso del operador “=”.



3.6.9.1 TraceEcommerce – Filtros gráfico lineal

Gráfico de barras

El primer campo para el filtro no actuará como tal, si no que se utiliza para agrupar los resultados, es decir, este campo será por el que se categoricen los resultados para mostrarse en el gráfico.



3.6.9.2 TraceEcommerce – Filtros gráfico de barras

No es necesario cambiar el operador ni introducir un valor porque no lo tomará en cuenta, sin embargo para los siguientes filtros ya si que se aplicarán como tal y habrá que darles valor.

Tabla

Introduciendo el campo y la condición que debe cumplir. Hay que tener en cuenta que para operar con cadenas solo será válido el uso del operador “=”.

date	hour	eventName	eventCategory	eventAction	eventLabel	eventValue
20200101	16:40:08	generic	checkout	gohome	-	-
20200101	16:41:10	generic	checkout	gohome	-	-
20200101	22:10:46	generic	checkout	gohome	-	-
20200101	22:11:14	generic	checkout	gohome	-	-
20200101	22:12:27	generic	checkout	gohome	-	-
20200102	09:08:40	generic	checkout	gohome	-	-
20200102	09:27:10	generic	checkout	gohome	-	-
20200103	16:12:16	generic	checkout	gohome	-	-

3.6.9.3 TraceEcommerce – Filtros gráfico tabla

4.6.1 Implementación

Para poder tener medición en el site es imprescindible hacer tener acceso a la función `_sendTo()` que se encargará de enviar la información recogida en la web a la base de datos de TFGdashboard.

Esta función debe ser editada por el usuario para establecer la cuenta y la vista a la que desea enviar los datos, para que haga una petición POST al archivo `insertDB.php` alojado en el servidor que será el que se encargue de hacer la inserción de la información en la base de datos.

Una de las ventajas de tener acceso a la función es que se puede seleccionar la cuenta y la vista a la que se van a enviar los datos desde el propio código sin tener que cambiar configuraciones desde tag manager o desde el DOM. El siguiente fragmento muestra cómo se puede elegir una vista dependiendo del entorno de la web:

```
function _sendTo(str){
    var count='1';
    var view='1';
    //////////////////////////////////////
    if(location.hostname.indexOf('.pre.')>-1){
        view='2';
    }else if(location.hostname.indexOf('.test.')>-1){
        view='3';
    }
    //////////////////////////////////////
    str['count']=count;
    str['view']=view;
    jQuery.post("https://tfdashboard.de/tfdashboard/insertDB.php", str);
}
```

3.6.10.1 TraceEcommerce – Código de seguimiento condicionado

Si es un entorno de preproducción enviará a la vista 2, si es el entorno de pruebas enviará a la vista 3, y si está en producción enviará a la vista 1. De este modo se evita el contaminar los datos de las vistas con pruebas y valores incoherentes.

Una desventaja que tiene es que no se puede enviar a dos vistas a la vez, lo que puede ser útil en caso de querer tener una prueba con tráfico real (en producción) aunque no se muy corriente realizar estas prácticas. Por último, hay que destacar que este script se puede introducir de tres maneras distintas:

Incluir el código en cada página

La ventaja es que es seguro que cargará siempre que cargue la página y con la mayor rapidez posible, sin embargo, hay que añadirlo a todas las páginas lo que es un trabajo pesado y si hubiera que hacer cualquier modificación habría que hacer el cambio en todas las páginas por lo que no sería muy práctico.

Incluir desde un archivo externo

La ventaja que tiene es que solo habría un archivo que es solicitado desde todas las páginas por lo que sería bastante eficiente cuando hubiera que hacer modificaciones y se podría cargar en cualquier punto del DOM, quedaría a elección del desarrollador, aunque se recomienda lo antes posible. Aunque seguiría existiendo el problema de que habría que añadir la llamada al script en todas las páginas al menos en la primera implementación y si hubiera un cambio de nombre o de ruta del archivo.

En mi caso opté por esta porque la página ya la tenía desarrollada con anterioridad y no tenía implementado ningún tag manager cuando hice el desarrollo.

Incluir desde el tag manager

Por último, esta opción considero que es la más completa ya que se puede llamar junto con la carga de página lo más pronto posible que permita el tag manager y si hay que hacer modificaciones solo habría que modificar el script en el tag manager. Además, no sería necesario llamarlo en página ya que se ejecutaría con la llamada al tag manager.

5 Integración, pruebas y resultados

5.1 Pruebas en página

Este tipo de pruebas consisten en confirmar que la recolección de datos y el comportamiento del core avcs es el correcto. Para ello me ayudé de la consola de herramientas para desarrolladores que ofrecen los navegadores, en mi caso usé la de Google Chrome.

Se comprobó que los pageviews y eventos se desarrollan correctamente mediante las funciones de impresión, y que en cada página que los valores recogidos por el core avcs son correctos y también que los objetos propios de las herramientas de analítica activas están correctamente formados como en caso de Google Analytics el dataLayer (Anexo I).

Estas pruebas permiten determinar si se están recogiendo correctamente los valores para cada página y si el flujo que sigue el core avcs para enviar los pageviews y eventos es el correcto.

Para hacer esta comprobación se ha usado el documento de valores de recolección comparando cada pageview y evento que aparece en el con el implementados en la página y apuntando si es correcto o incorrecto.

Los valores incorrectos se han corregido tras cada revisión.

5.2 Pruebas en Google Tag Manager

Las pruebas que se han realizado desde el tag manager consisten en confirmar el correcto funcionamiento de las etiquetas, triggers y variables implementados en Google Tag Manager y que se producen los envíos a las distintas herramientas de analítica, en este caso Google Analytics y TFGdashboard.

Para probar el funcionamiento de Google Tag Manager se han realizado test según se iba haciendo la implementación por lo que no ha habido varias revisiones, y se ha utilizado la funcionalidad de vista previa disponible en Google Tag Manager, donde se ha comprobado que la configuración es la correcta, el flujo funciona y las variables que se enviarán a Google Analytics están correctamente alimentadas (Anexo J).

Por otro lado se ha comprobado que los hits de pageviews y eventos son enviados a las herramientas de analítica correspondientes con los valores correctos utilizando en las herramientas para desarrolladores la funcionalidad de network donde se pueden ver las peticiones HTTP de la página.

Para TFGdashboard, se puede encontrar filtrando por **insertDB.php** (Anexo H).

Como a Google Analytics, se puede encontrar filtrando por **collect** y viendo el contenido de la traza porque a veces aparecen otras peticiones como los timings que son automáticos. De todas formas el contenido en el resultados es importante para identificar la traza que queremos identificar (Anexo H). Con el enhanced ecommerce incluido, también se puede encontrar filtrando por **collect** y fijándonos en el contenido (Anexo H).

5.3 Pruebas en informes

La última prueba es comprobar en las herramientas de analítica que los datos están llegando. Para ello he realizado distintas pruebas dependiendo la herramienta.

TFGdashboard

He comprobado diariamente que los datos llegan y hay tráfico en general y también se han realizado pruebas con valores específicos para poder identificar los datos en la herramienta.

También se hicieron pruebas de eventos aislados con valores específicos que se comprobaron en la base de datos para confirmar que efectivamente se estaban almacenando. Estas pruebas se borraron de la base de datos para que no apareciesen en la herramienta y contaminaran los datos.

Google Analytics

Las pruebas que se han realizado para esta herramienta son muy similares a las que se hicieron para TFGdashboard pero con la excepción de que aquí no hay acceso a los datos en bruto (base de datos) y todo lo que se envíe quedará registrado.

De todas formas se comprobó que hay un tráfico constante de datos, y para más seguridad se utilizó la funcionalidad del **real time** y una url parametrizada para comprobar en tiempo real que la navegación que se hace se está registrando.

También se comprobaron que todas las acciones de enhanced ecommerce se registraron correctamente en la herramienta haciendo especial énfasis en las compras.

Aquí encontré el problema de que las pruebas quedan también registradas y no se pueden borrar, por lo que hay compras con cantidades excesivas de productos o con precios muy altos que durante el periodo de pruebas distorsionan los datos reales y que no se pueden eliminar.

Una opción del para el futuro puede ser utilizar una propiedad para enviar los datos cuando se estén haciendo cualquiera de las pruebas anteriores incluidas las de Google Tag Manager.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Considero que la solución desarrollada durante este trabajo puede ser una buena oportunidad para que páginas web y negocios que no se han planteado todavía implementar soluciones para el seguimiento en sus webs.

Hoy en día son pocos los negocios que no poseen una página web a través de la cual ofrecen sus servicios y tener un buen plan de analítica implementada en ellas esencial para conocer que rendimiento está dando la web y como está ayudando a un negocio llegando a poder ser una parte vital en el crecimiento del mismo.

La implementación de la solución expuesta supondría un trabajo bastante asequible, unos tiempos de implementación realmente cortos y resultados inmediatos. En realmente poco tiempo se tendría una medición de un site web bastante completo y se podría empezar a hacer estudios y obtener resultados de métricas tan importantes como cuanto tiempo pasan los usuarios en la página web, si hay algún paso del proceso de compra donde los usuarios se salen y cuál puede ser la razón o cuales son los canales de entrada más habituales a la web.

Además de poder ser un buen primer paso en la analítica también puede suponer una mejora para sites que ya tienen una solución implementada pero no está muy cuidada o necesitan una actualización ya que este SDK ofrece sencillez, escalabilidad y mantenibilidad por lo cual puede ser una opción bastante atractiva.

Personalmente me ha ayudado a comprender las necesidades principales y más útiles que pueden tener los sitios web para medirlos e informarlos a plataformas de analítica donde se puedan hacer estudios de relevancia para su mejora y crecimiento, además de poner en valor la recolección de estos datos ya sea extrayéndolos directamente del DOM o siendo informados por el equipo de desarrolladores en la propia página. También me ha permitido mejorar mi conocimiento en herramientas de analítica y tag manager como son Google

Analytics y Google Tag Manager que son fundamentales para la medición de las páginas web.

Por último, quiero destacar que la creación de una herramienta de analítica como TFGdashboard me ha permitido plantear cómo funciona el flujo y la integración entre web, tag manager y herramientas de analítica, un aspecto fundamental cuando se quieren sacar el máximo de datos disponibles en un site y comprender las variante o casuísticas que pueden darse desde que se obtiene el dato hasta que se puede visualizar de una forma más amigable.

6.2 Trabajo futuro

Debido a la estructura con la que se planificó el SDK está en constante evolución y mejora de sus funcionalidades por lo tanto depende completamente de las necesidades de la web donde se implemente. Sin embargo, hay ciertas funcionalidades básicas que creo que podrían añadirse como una librería para facilitar el manejo de cookies, una librería para gestionar localStorage y sessionStorage con fechas, o implementar directamente el módulo del enhanced ecommerce aprovechando la propia estructura del SDK se podría construir en él mismo.

Además, cabe destacar que el mundo de la analítica web está en constante movimiento y cambio lo cuál de por sí obligará a tener que cambiarlo y añadirle nuevas funcionalidades que irán apareciendo de la mano de la evolución de las páginas web, las nuevas herramientas que aparezcan en el mercado y las necesidades de los clientes.

En cuanto a las posibles mejoras de la plataforma TFGdashboard son bastante más claras ya que en principio no se concibió como una herramienta con un gran alcance si más bien como una herramienta para la visualización básica de los datos recogidos, para comprender como otras herramientas podían deducir ciertas métricas o estudiar porque son importantes, y para comprender el flujo de datos entre su recolección y si visualización.

Aun así, considero que se podría mejorar el sistema de visualización de las tablas indicando solo las métricas que se quieren visualizar y no todas las disponibles. También se podría mejorar el tipo de eventos enfocado solo en el estudio de variables de compra (ecommerce) de productos similar al enhanced ecommerce de Google, pero creo que con algunas mejoras como más campos y añadiendo un sistema para gestionar las muestras o regalos que a veces vienen con las compras y que el enhanced ecommerce no considera.

También se podría añadir un módulo avanzado para gestionar campañas, algo que es muy importante en los sites web pero que de momento no puedo implementar por desconocimiento del funcionamiento tanto de las campañas como de cómo deben gestionarse por la herramienta.

Referencias

- [1] Medición del comercio electrónico mejorado
(<https://developers.google.com/analytics/devguides/collection/analyticsjs/enhanced-ecommerce?hl=es-419>)
- [2] Documentación Google Tag Manger
(<https://support.google.com/tagmanager/answer/6102821?hl=es>)
- [3] Variables en Google Tag Manager (<https://www.simoahava.com/analytics/variable-guide-google-tag-manager/#:~:text=In%20Google%20Tag%20Manager%2C%20the,explored%20in%20the%20previous%20paragraph.>)
- [4] Triggers/Activadores Google Tag Manger
(<https://aukera.es/blog/guia-activadores-google-tag-manager/>)
- [5] Tags/Etiquetas Google Tag Manager
(<https://support.google.com/tagmanager/answer/3281060?hl=es>)
- [6] Documentación de Query (<https://api.jquery.com/>)
- [7] Documentación de Google Analytics
(<https://rockcontent.com/es/blog/google-analytics/>)
- [8] Propiedades y vistas Google Analytics
(<https://support.google.com/analytics/answer/1009618?hl=es>)
- [9] Propiedades web+app en Google Analytics (<https://www.analiticaweb.es/google-analytics-app-web/>)
- [10] Servidores Hetzner (<https://www.hetzner.com/>)
- [11] Firebase (<https://openwebinars.net/blog/que-es-firebase-de-google/>)
- [12] Firebase login e identificación de usuarios (<https://firebase.google.com/docs/auth>)
- [13] MySQL (<https://www.mysql.com/>)
- [14] Documentación de Plotly.js (<https://plotly.com/>)
- [15] Agrupación de canales (<https://imorillas.com/analytics/canales-google-analytics/>)
- [16] Adwords (https://ads.google.com/intl/es_es/home/)
- [17] Google Campaign Manager
(<https://support.google.com/dcm/answer/2709362?hl=es>)
- [18] Gestión de cookies en JavaScript
(<https://developer.mozilla.org/es/docs/DOM/document.cookie>)
- [19] Utilización de storage (<https://ed.team/blog/que-es-y-como-utilizar-localstorage-y-sessionstorage>)

Glosario

- Analítica web: la medición, procesamiento, análisis y reporte del tráfico de internet para entender y optimizar el uso de un sitio web
- Evento: son interacciones del usuario con contenido que se pueden medir de forma independiente a partir de una página web o una carga de pantalla
- GA: es un servicio gratuito prestado por Google que ofrece un gran conjunto de estadísticas de tu página/s web/s
- GTM: es el tag manager desarrollado por Google.
- Marketing digital: son todas aquellas acciones y estrategias publicitarias o comerciales que se ejecutan en los medios y canales de internet.
- Pageview: Número de páginas vistas que se cargan (o se vuelven a cargar) en un navegador.
- SDK (software development kit): es un paquete de herramientas y datos que facilita e incluso permite a los programadores desarrollar programas en un lenguaje concreto o para una plataforma o aplicación específica.
- Sesión: Una sesión es un conjunto de interacciones que tienen lugar en su sitio web desde que se accede a la web hasta que se cierra la web o el navegador.
- Site: es un conjunto de páginas web relacionadas a un dominio de internet.
- Tag manager: es un sistema de administración o gestor de etiquetas que nos permite actualizar de forma fácil y rápida todas aquellas etiquetas/píxeles/códigos dentro de su plataforma sin tener que añadir directamente el código en la web o apps.
- TFGdashboard: herramienta desarrollada durante este proyecto para visualizar y analizar los datos en enviados desde el core avcs.

Anexos

A Manual de instalación

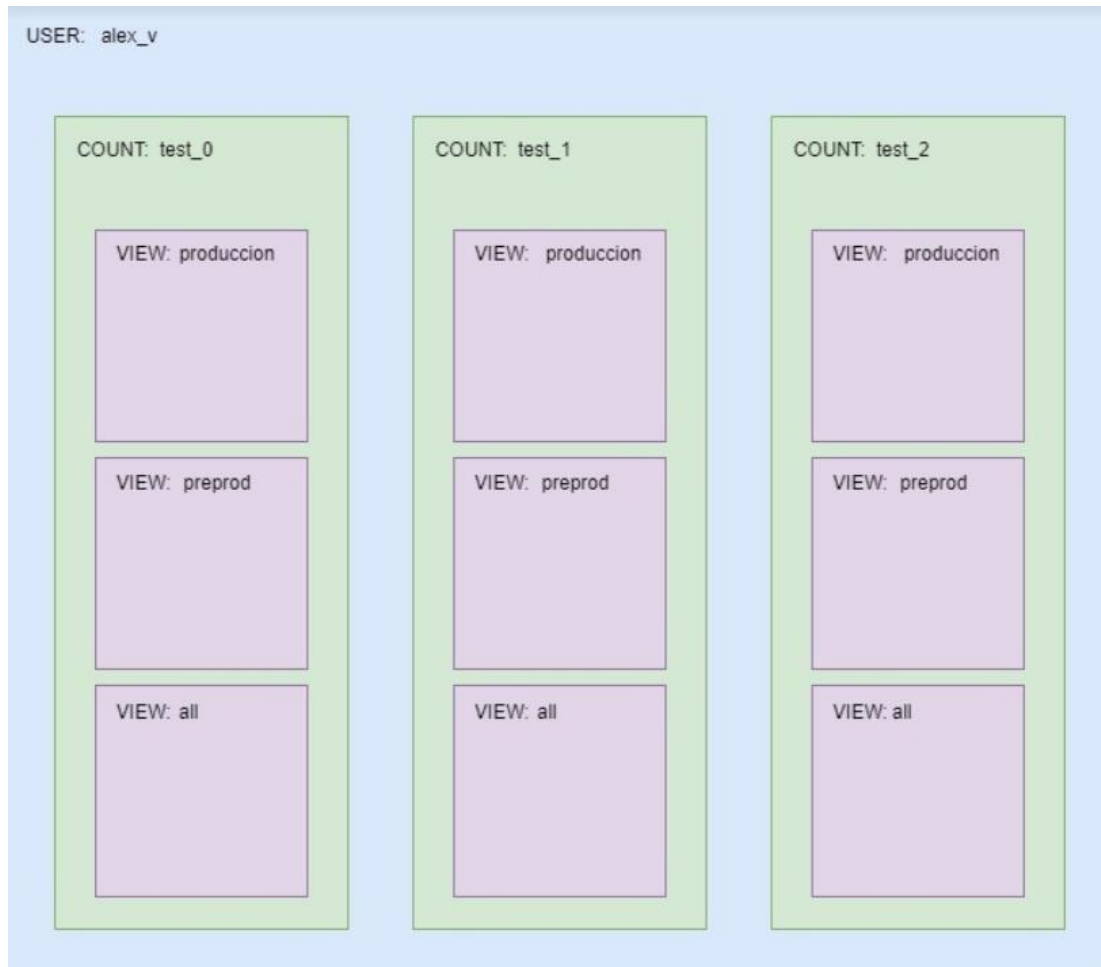
1. Crear una cuenta en Google Tag Manager y Google Analytics (no es obligatorio, pero si es recomendable para tener un apoyo) en modo web.
2. Hacer un estudio de la página y generar un documento de Valores de recolección (Anexo C)
3. Añadir referencia a la función `_sendTo()`
4. Incluir el core avcs en Google Tag Manager y ajustarlo a las necesidades de la página.
 - a. Seleccionar herramientas de analítica activas
 - b. Asignar valores de recolección adecuados
 - c. Añadir funciones necesarias
5. Crear variables en Google Tag Manager
6. Configurar tags y triggers en Google Tag Manager para gestionar eventos.
7. Activar opción de enhanced ecommerce en Google Analytics si se requiere
8. Añadir tags opcionales para TFGdashboard si se considera oportuno
9. Crear un usuario en TFGdashboard
10. Crear una cuenta y vista (al menos) en TFGdashboard asociada al usuario
11. Configurar el contenido de la función `_sendTo()` para enviar los datos a la propiedad y vista correspondientes
12. Comprobar en las herramientas de analítica implementadas (Google Analytics y TFGdashbaord) que los datos son correctos.

Para cualquier duda en los apartados 3 y 4 se entra más en detalle en como se debe implementar cada punto y su funcionamiento.

B Documento de recolección para páginas

			Página			Página			Página		
			https://tstodashboard.delecommerceTestmanuel.htm			https://tstodashboard.delecommerceTestproduct.htm			https://tstodashboard.delecommerceTestcart.htm		
Hit	URL		Página	Página	Página	Página	Página	Página	Página	Página	Página
Descripción			https://tstodashboard.delecommerceTestmanuel.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm	https://tstodashboard.delecommerceTestproduct.htm
subcategoria1	3	P	Página de inicio								
subcategoria2	10	P									
subcategoria3	11	P									
subcategoria4	-	P									
date	12	PEWEC	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>	<YYYYMMDD>
weekday	13	P	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>	<lunesmartes,,domingo>
time	14	PEWEC	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>	<HHMMSS>
adblock	15	P	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>	<enabledisabled>
USER											
gaclient	-	P	-	-	-	-	-	-	-	-	-
useragent	16	P	<useragent>	<useragent>	<useragent>	<useragent>	<useragent>	<useragent>	<useragent>	<useragent>	<useragent>
newsletter	-	P	-	-	-	-	-	-	-	-	-
fingerprint	-	P	-	-	-	-	-	-	-	-	-
idom	-	P	-	-	-	-	-	-	-	-	-
cookie	17	P	-	-	-	-	-	-	-	-	-
TRAFFIC											
campaign	-	P	-	-	-	-	-	-	-	-	-
source	-	P	-	-	-	-	-	-	-	-	-
medium	-	P	-	-	-	-	-	-	-	-	-
term	-	P	-	-	-	-	-	-	-	-	-
content	-	P	-	-	-	-	-	-	-	-	-
SEARCH											
keyword	18	P	-	-	-	-	-	-	-	-	-
resultnumber	19	P	-	-	-	-	-	-	-	-	-
originpage	20	P	-	-	-	-	-	-	-	-	-
DATAQUALITY											
qtrversion	-	-	-	-	-	-	-	-	-	-	-
trqversion	-	-	<trq version>	<trq version>	<trq version>	<trq version>	<trq version>	<trq version>	<trq version>	<trq version>	<trq version>
PRODUCTS											
name	-	-	-	-	-	-	-	-	-	-	-
id	-	EC	-	-	-	-	-	-	-	-	-
price	-	-	-	-	-	-	-	-	-	-	-
brand	-	-	-	-	-	-	-	-	-	-	-
category	-	-	-	-	-	-	-	-	-	-	-
variant	-	-	-	-	-	-	-	-	-	-	-
list_position	-	-	-	-	-	-	-	-	-	-	-
PROMOTIONS											
id	-	-	-	-	-	-	-	-	-	-	-
name	-	-	-	-	-	-	-	-	-	-	-
creative	-	-	-	-	-	-	-	-	-	-	-
position	-	-	-	-	-	-	-	-	-	-	-
ECOMMERCE											
transactionid	-	EC	-	-	-	-	-	-	-	-	-
price	-	EC	-	-	-	-	-	-	-	-	-
step	-	EC	-	-	-	-	-	-	-	-	-
quantity	-	EC	-	-	-	-	-	-	-	-	-
shipping	-	EC	-	-	-	-	-	-	-	-	-
discountValue	-	EC	-	-	-	-	-	-	-	-	-
discountCode	-	EC	-	-	-	-	-	-	-	-	-
tax	-	EC	-	-	-	-	-	-	-	-	-
CUSTOM											
-	-	-	-	-	-	-	-	-	-	-	-
EVENTS											
eventName (event)	-	EV	-	-	-	-	-	-	-	-	-
eventCategory	-	EWEC	-	-	-	-	-	-	-	-	-
eventAction	-	EWEC	-	-	-	-	-	-	-	-	-
eventLabel	-	EWEC	-	-	-	-	-	-	-	-	-

C Jerarquía de usuarios en TFGdashboard



D Documento de recolección para eventos

Hit				Evento			
URL				-	-	-	-
Descripción				Se lanza al clicar sobre la casa para volver a la home			
subcategorí1	9	P		-	-	-	-
subcategorí2	10	P		-	-	-	-
subcategorí3	11	P		-	-	-	-
subcategorí4	-	P		-	-	-	-
date	12	PIE/EC		-	-	-	-
weekday	13	P		-	-	-	-
time	14	PIE/EC		-	-	-	-
adblock	15	P		-	-	-	-
USER							
gaclient	-	P		-	-	-	-
useragent	16	P		-	-	-	-
newsletter	-	P		-	-	-	-
fingerprint	-	P		-	-	-	-
idcm	-	P		-	-	-	-
cookie	17	P		-	-	-	-
TRAFFIC							
campaign	-	P		-	-	-	-
source	-	P		-	-	-	-
medium	-	P		-	-	-	-
term	-	P		-	-	-	-
content	-	P		-	-	-	-
SEARCH							
keyword	18	P		-	-	-	-
resultnumber	19	P		-	-	-	-
originpage	20	P		-	-	-	-
DATAPROBILITY							
dmversion	-	-		-	-	-	-
ttversion	-	-		-	-	-	-
PRODUCTS							
name	-	-		-	-	-	-
id	-	EC		-	-	-	-
price	-	-		-	-	-	-
brand	-	-		-	-	-	-
category	-	-		-	-	-	-
variant	-	-		-	-	-	-
list_position	-	-		-	-	-	-
PROMOTIONS							
id	-	-		-	-	-	-
name	-	-		-	-	-	-
creative	-	-		-	-	-	-
position	-	-		-	-	-	-
ECOMMERCE							
transactionid	-	EC		-	-	-	-
price	-	EC		-	-	-	-
step	-	EC		-	-	-	-
quantity	-	EC		-	-	-	-
shipping	-	EC		-	-	-	-
discountValue	-	EC		-	-	-	-
discountCode	-	EC		-	-	-	-
tax	-	EC		-	-	-	-
CUSTOM							
-	-	-		-	-	-	-
EVENTS							
eventName (event)	-	EV		genero	genero	genero	genero
eventCategory	-	EW/EC		userAction	checkout	checkout	checkout
eventAction	-	EW/EC		header	start checkout	pay	shipping
eventLabel	-	EW/EC		goHome	-	<pay method>	-

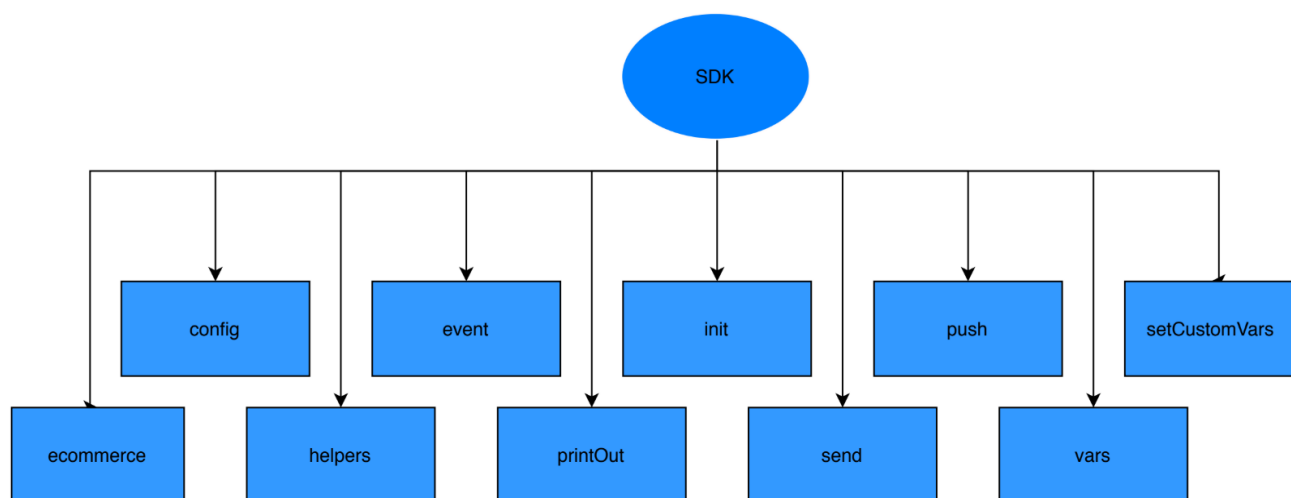
E Documento de recolección para eventos de ecommerce

				Evento	Evento	Evento	Evento	Evento	
	Hit URL			Registra transacciones	Producto añadido al carrito	Producto eliminado del carrito	Pulsar pagar en paso 1 de compra	Pulsar comprar en paso 2 de compra	
subcateqon1	9	P		-	-	-	-	-	-
subcateqon2	10	P		-	-	-	-	-	-
subcateqon3	11	P		-	-	-	-	-	-
subcateqon4	-	P		-	-	-	-	-	-
date	12	PIEWEC		-	-	-	-	-	-
weekday	13	P		-	-	-	-	-	-
time	14	PIEWEC		-	-	-	-	-	-
abblock	15	P		-	-	-	-	-	-
USER									
qualient	-	P		-	-	-	-	-	-
useridant	16	P		-	-	-	-	-	-
newsletter	-	P		-	-	-	-	-	-
fingerprint	-	P		-	-	-	-	-	-
idom	-	P		-	-	-	-	-	-
cookie	17	P		-	-	-	-	-	-
TRAFFIC									
campaign	-	P		-	-	-	-	-	-
source	-	P		-	-	-	-	-	-
medium	-	P		-	-	-	-	-	-
term	-	P		-	-	-	-	-	-
content	-	P		-	-	-	-	-	-
SEARCH									
keyword	18	P		-	-	-	-	-	-
resultnumber	19	P		-	-	-	-	-	-
originpage	20	P		-	-	-	-	-	-
QUALITY									
conversion	-	-		-	-	-	-	-	-
trqversion	-	-		-	-	-	-	-	-
PRODUCTS									
name	-	-		-	<name>	<name>	-	-	-
id	-	EC		-	<ids>	<ids>	-	-	-
price	-	-		-	<price>	<price>	-	-	-
brand	-	-		-	<brand>	<brand>	-	-	-
cateqon	-	-		-	<cateqon>	<cateqon>	-	-	-
variant	-	-		-	<variant>	<variant>	-	-	-
list position	-	-		-	<list position>	<list position>	-	-	-
PROMOTIONS									
id	-	-		-	-	-	-	-	-
name	-	-		-	-	-	-	-	-
creative	-	-		-	-	-	-	-	-
position	-	-		-	-	-	-	-	-
ECCOMMERCE									
transactionid	-	EC		-	-	-	-	-	-
price	-	EC		-	<price>	<price>	-	-	-
step	-	EC		-	<step>	<step>	-	-	-
quantity	-	EC		-	<quantity>	<quantity>	-	-	-
shipping	-	EC		-	<shipping price>	<shipping price>	-	-	-
discountValue	-	EC		-	<discount>	<discount>	-	-	-
discountCode	-	EC		-	<discount code>	<discount code>	-	-	-
tax	-	EC		-	<tax>	<tax>	-	-	-
CUSTOM									
-	-	-		-	-	-	-	-	-
EVENTS									
eventName (event)	-	EV		ecommerce transaction	ecommerce add to cart	ecommerce remove from cart	ecommerce pay	ecommerce shipping	
eventCategory	-	EWEC		-	-	-	-	-	-
eventLabel	-	EWEC		-	-	-	<pay method>	-	-

F Documento de recolección de eventos de eventos para GA




Hit			Evento									
URL			Evento	Evento	Evento	Evento	Evento	Evento	Evento	Evento		
Descripción			-	-	-	-	-	-	-	-	-	-
subcategory1	9	P	-	-	-	-	-	-	-	-	-	-
subcategory2	10	P	-	-	-	-	-	-	-	-	-	-
subcategory3	11	P	-	-	-	-	-	-	-	-	-	-
subcategory4	-	P	-	-	-	-	-	-	-	-	-	-
date	12	PIVEEC	-	-	-	-	-	-	-	-	-	-
weekday	13	P	-	-	-	-	-	-	-	-	-	-
time	14	PIVEEC	-	-	-	-	-	-	-	-	-	-
adblock	15	P	-	-	-	-	-	-	-	-	-	-
USER												
gaclient	-	P	-	-	-	-	-	-	-	-	-	-
useragent	16	P	-	-	-	-	-	-	-	-	-	-
newsletter	-	P	-	-	-	-	-	-	-	-	-	-
fingerprint	-	P	-	-	-	-	-	-	-	-	-	-
idorm	-	P	-	-	-	-	-	-	-	-	-	-
cookie	17	P	-	-	-	-	-	-	-	-	-	-
TRAFFIC												
campaign	-	P	-	-	-	-	-	-	-	-	-	-
source	-	P	-	-	-	-	-	-	-	-	-	-
medium	-	P	-	-	-	-	-	-	-	-	-	-
term	-	P	-	-	-	-	-	-	-	-	-	-
content	-	P	-	-	-	-	-	-	-	-	-	-
SEARCH												
keyword	18	P	-	-	-	-	-	-	-	-	-	-
resultnumber	19	P	-	-	-	-	-	-	-	-	-	-
ordidpage	20	P	-	-	-	-	-	-	-	-	-	-
DATAQUALITY												
qtmversion	-	-	-	-	-	-	-	-	-	-	-	-
qtversion	-	-	-	-	-	-	-	-	-	-	-	-
PRODUCTS												
name	-	-	<name>									
id	-	EC	<id>									
price	-	-	<price>									
brand	-	-	<brand>									
category	-	-	<category>									
variant	-	-	<variant>									
list position	-	-	<list position>									
PROMOTIONS												
id	-	-										
name	-	-										
creative	-	-										
position	-	-										
ECOMMERCE												
transactionid	-	EC										
price	-	EC										
step	-	EC										
quantity	-	EC										
shipping	-	EC										
discountValue	-	EC										
discountCode	-	EC										
tax	-	EC										
CUSTOM												
-	-	-										
EVENTS												
eventName (event)	-	EV										
eventCategory	-	EVIEC										
eventAction	-	EVIEC										
eventLabel	-	EVIEC										
			generic engagement view_item_list	generic engagement select_content	generic engagement view_item	generic ecommerce add_to_cart	generic ecommerce remove_from_cart	generic ecommerce checkout_progress	generic ecommerce purchase			

G Esquema avcs





H Trazas de debug

Traza de TFGdashboard:

 insertDB.php /tfgdashboard	content[referrer]:
 ec.js www.google-a...	content[webmode]: mobile
 collect?v=1&_v=j... www.google-a...	content[language]: es-es
	content[date]: 20200701
	content[weekday]: miercoles
	content[time]: 23:07:25
	content[adblock]: enabled
	user[gaclient]: ga1.2.642200100.1590327832
	user[useragent]: mozilla/5.0 (windows nt 10.0; win64; x64) applewebkit/537.36 (khtml, like gecko) chrome/83.0.4103.116 safari/537.36
	dataquality[tfversion]: 2.1
	products[0][id]: 00000001
	products[0][name]: camiseta azul
	products[0][price]: 10
	products[0][category]: camiseta
	products[1][id]: 00000002
	products[1][name]: camiseta gris
	products[1][price]: 18
	products[1][category]: camiseta
	products[2][id]: 00000003
	products[2][name]: botas marrones
	products[2][price]: 40
	products[2][category]: calzado
	products[3][id]: 00000004
	products[3][name]: zapatillas azules
	products[3][price]: 25
	products[3][category]: calzado
	events[0][event]: pageview
	userId: ga1.2.642200100.1590327832
	event: pageview
	count: 10
	view: 3



1 / 31 requests | 23 B /

Traza de GA:

Name	X	Headers	Preview	Response	Initiator	Timing
 collect?v=1&_v=j... www.google-anal...	t: pageview					
 collect?v=1&_v=j... www.google-anal...	dl: https://tfgdashboard.de//ecommerceTest/menu.php					
	ul: es-es					
	de: UTF-8					
	dt: Inicio					
	sd: 24-bit					
	sr: 1920x1080					
	vp: 626x937					
	je: 0					
	_u: SCCAAEAL~					
	jid: 395965466					
	gjid: 357029825					
	cid: 642200100.1590327832					
	tid: UA-158350460-1					
	_gid: 2011209127.1593636442					
	_r: 1					
	gtm: 2wg6o0M8VH58					
	cd1: home					
	cd2: home					
	cd3: tfgdashboard.de					
	cd4: https://tfgdashboard.de//ecommercetest/menu.php					
	cd5:					
	cd6: mobile					
	cd7: es-es					
	cd12: 20200701					
	cd13: miercoles					
	cd14: 23:07:25					
	cd15: enabled					
	cd16: mozilla/5.0 (windows nt 10.0; win64; x64) applewebkit/537.36 (khtml, like gecko) chrome/83.0.4103.116 safari/537.36					
	cd17: 49105864050832300					

2 / 31 requests | 195 B /

Traza de Enhanced Ecommerce:

Name	×	Headers	Preview	Response	Initiator	Timing
 collect?v=1&_v=j... www.google-anal...		dt: Inicio sd: 24-bit sr: 1920x1080 vp: 609x937 je: 0 ec: engagement ea: view_item_list _u: SCCAAEAL~ jid: gjid: cid: 642200100.1590327832 tid: UA-158350460-1 _gid: 2011209127.1593636442 gtm: 2wg6o0MN8VH58 il1pi1id: 00000001 il1pi1nm: camiseta azul il1pi1pr: 10 il1pi1ca: camiseta il1pi2id: 00000002 il1pi2nm: camiseta gris il1pi2pr: 18 il1pi2ca: camiseta il1pi3id: 00000003 il1pi3nm: botas marrones il1pi3pr: 40 il1pi3ca: calzado il1pi4id: 00000004 il1pi4nm: zapatillas azules il1pi4pr: 25 il1pi4ca: calzado z: 518653812				
 collect?v=1&_v=j... www.google-anal...						

2 / 31 requests | 195 B /

I Valores de objetos en consola

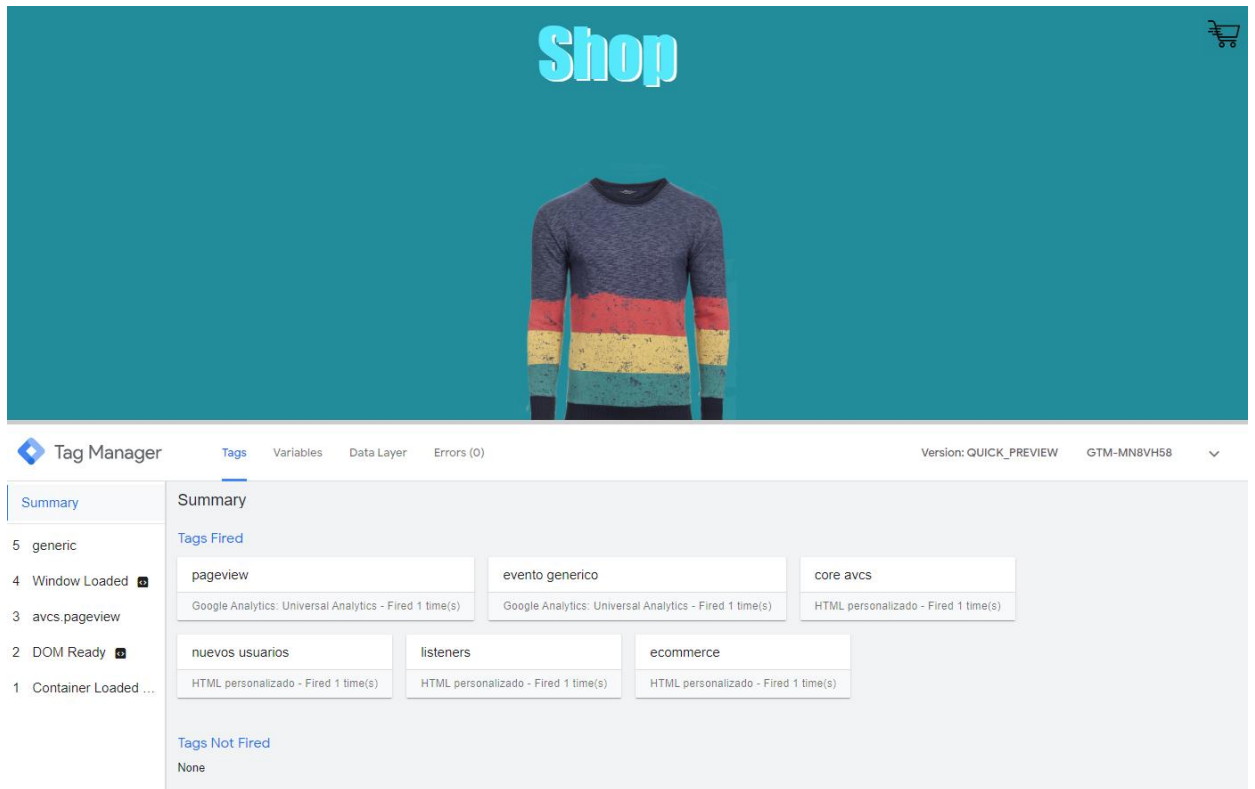
Valore de avcs:

```
> avcs
< ▼ {config: {...}, event: "", vars: {...}, helpers: {...}, printOut: f, ...} ⓘ
  ▶ config: {version: "3.0", tms: Array(5), ecommerce: false, printEnable: 0, styles: Array(4)}
  ▶ ecommerce: f ()
  ▶ event: ""
  ▶ helpers: {ceros: f, getTime: f, truevalue: f, getParam: f, getGAcookie: f, ...}
  ▶ init: f ()
  ▶ printOut: f (a,b,c)
  ▶ push: f (a)
  ▶ send: f (a)
  ▶ setCustomVars: f ()
  ▶ vars: {content: {...}, user: {...}, traffic: {...}, search: {...}, dataquality: {...}, ...}
  ▶ __proto__: Object
```

Valor de dataLayer:

```
dataLayer
▼ (5) [{...}, {...}, {...}, {...}, {...}, push: f] ⓘ
  ▶ 0: {gtm.start: 1593636460404, event: "gtm.js", gtm.uniqueEventId: 0}
  ▶ 1: {event: "gtm.dom", gtm.uniqueEventId: 1}
  ▶ 2: {content: {...}, user: {...}, traffic: {...}, search: {...}, dataquality: {...}, ...}
  ▶ 3: {event: "gtm.load", gtm.uniqueEventId: 4}
  ▶ 4: {event: "generic", eventCategory: "engagement", eventAction: "view_item_list", eventLabel: undefined, ecommerce: {...}, ...}
  ▶ push: f ()
  ▶ length: 5
  ▶ __proto__: Array(0)
```

J Vista previa GTM (Preview mode)



The screenshot displays the Google Tag Manager (GTM) preview mode interface. At the top, a teal banner features the word "Shop" in a large, light blue, stylized font. Below the banner is a product image of a long-sleeved sweater with a dark blue body and horizontal stripes in red, yellow, and green. In the top right corner of the banner, there is a small shopping cart icon.

Below the banner is the GTM interface. The top navigation bar includes the "Tag Manager" logo, tabs for "Tags", "Variables", "Data Layer", and "Errors (0)", and version information: "Version: QUICK_PREVIEW" and "GTM-MN8VH58".

The left sidebar shows a list of tags with their execution order:

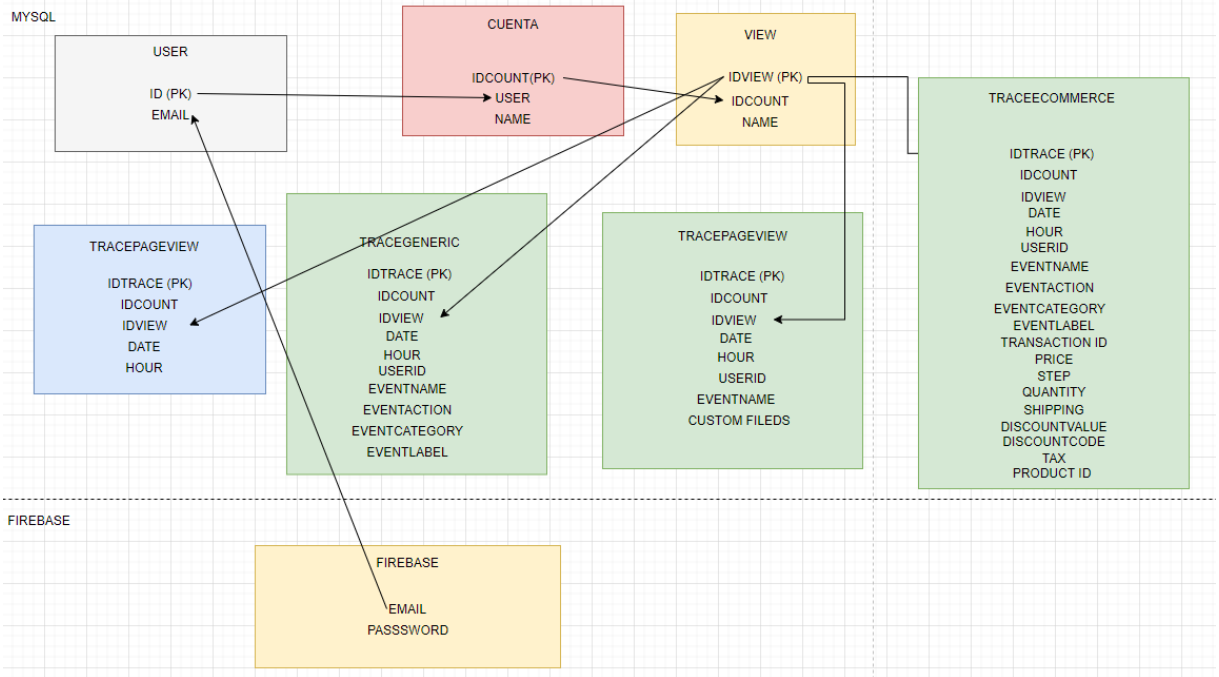
- 5 generic
- 4 Window Loaded
- 3 avcs.pageview
- 2 DOM Ready
- 1 Container Loaded...

The main content area is titled "Summary" and shows the "Tags Fired" section. It contains six tag boxes, each with its name and the number of times it has fired:

Tag Name	Fired Count
pageview	Google Analytics: Universal Analytics - Fired 1 time(s)
evento generico	Google Analytics: Universal Analytics - Fired 1 time(s)
core avcs	HTML personalizado - Fired 1 time(s)
nuevos usuarios	HTML personalizado - Fired 1 time(s)
listeners	HTML personalizado - Fired 1 time(s)
ecommerce	HTML personalizado - Fired 1 time(s)

Below the "Tags Fired" section, the "Tags Not Fired" section is shown, indicating "None".

K Esquema base de datos



L Acciones de enhanced ecommerce

Impresiones de producto:

```
dataLayer.push({
  'ecommerce': {
    'currencyCode': 'EUR',
    'impressions': [
      {
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
        'list': 'New shoes',
        'position': 1
      }
    ]
  }
});
```

Detalle de productos:

```
dataLayer.push({
  'ecommerce': {
    'detail': {
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
      }]
    }
  }
});
```

Clicks en producto:

```
dataLayer.push({
  'ecommerce': {
    'click': {
      "actionField": {
        "list": "product list"
      },
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
        'position': 1
      }]
    }
  }
});
```

Pasos del checkout:

```
dataLayer.push({
  'ecommerce': {
    'checkout': {
      "actionField": {
        'step': 1
      },
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
      }]
    }
  }
});
```

Transacciones:

```
dataLayer.push({
  'ecommerce': {
    "currencyCode": "EUR",
    'purchase': {
      "actionField": {
        "id": 1,
        "affiliation": "ecommerceTest",
        "revenue": '20.50',
        "tax": '5.25',
        "shipping": '0'
      },
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
      }]
    }
  }
});
```

Añadir producto al carrito:

```
dataLayer.push({
  'ecommerce': {
    "currencyCode": "EUR",
    'add': {
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
        'quantity': 2
      }]
    }
  }
});
```

Eliminar producto del carrito:

```
dataLayer.push({
  'ecommerce': {
    "currencyCode": "EUR",
    'remove': {
      'products': [{
        'name': 'Product1',
        'id': '12345',
        'price': '15.25',
        'brand': 'EcommerceTest',
        'category': 'Shoes',
        'variant': 'Black',
        'quantity': 2
      }]
    }
  }
});
```


M Código de seguimiento GTM

En el head:

```
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-MN8VH58');
<!-- End Google Tag Manager -->
```

En el body:

```
<!-- Google Tag Manager (noscript) -->
<noscript><iframe src="https://www.googletagmanager.com/ns.html?id=GTM-MN8VH58"
height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
<!-- End Google Tag Manager (noscript) -->
```